
6,- DM öS 50,- sfr 6,-
CPC Amstrad
INTERNATIONAL
CPC • PCW JOYCE
12/1
Dezember 1991/
Januar 1992
8. Jahrgang
Erste Hilfe
Tips für Spielefreunde
Programmiersprachen
– Basic-Kurs: Weg mit dem GOTO
– Vier C-Compiler im Test
– Für Assemblerfreunde: RSX-Befehle »selbstgestrickt«
– Register im Griff: Inline-Hilfe für Turbo Pascal
Große RSX-Bibliothek
– Alles, was Ihrem BASIC immer schon gefehlt hat
– 35 neue Befehle
– Pulldown-Menüs, RAM-Disk, Interruptverwaltung und, und, und ...
Kniffe und Kunstgriffe
– Binärdateien verschlüsseln
– Sortier-Algorithmen
– Widerstands-Kombinationen ermitteln
– Grafik-Konvertierung unter CP/M
PCW
– Grafiksystem im Eigenbau
– Neue LocoScript-Version
– Top-Software: Programmiertes Lexikon
Der Neue
PCW 512 plus
Brandaktuelle Spiele aus Frankreich

DMV Überraschungspaket für CPC

Jetzt zugreifen!

Programme, Spiele, Bücher zum Wahnsinnspreis

Total verrückt! Software zum Leer-Disketten/Cassetten-Preis

Das Super-Paket zum Spar-Preis

In jedem Paket finden Sie:

10 Disketten oder Cassetten mit Top-Programmen und Spitzen-Spielen
+ **2** DATA BECKER-Bücher

3"-Disketten-Paket nur 49,—*

Cassetten-Paket nur 29,—*

Sofort bestellen! Nur solange Vorrat reicht.

* Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland DM 4,– bzw. für das Ausland DM 6,– Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.

DMV-Verlag • Postfach 250 • 3440 Eschwege





Liebe Leserinnen und Leser,

was würden Sie denken, wenn Ihnen an der nächsten Kreuzung plötzlich ein lebendiger Dinosaurier über den Weg laufen und Sie zu allem Überfluß auch noch freundlich grüßen würde? So ähnlich dürfte es manchem EDV-Kundigen unserer Tage gehen, wenn man ihn mit unseren Programmen in Locomotive- oder Mallard-BASIC konfrontieren würde.
Nun sind die smarten Yuppies der neueren PC-Generation, die so gern mit Megahertz und Megabytes um sich werfen (wetten, daß "Mega" einer der ganz großen Mode-Mädchennamen dieses Jahrzehnts wird?), ohnehin nicht allzugut auf das "Herumfummeln mit Programmcode" zu sprechen. Und wenn es denn schon Programmierung sein muß, dann wenigstens gleich ein Compiler-System mit allen Schikanen: konfigurierbare Oberfläche, am besten mit Icons und Fensterchen, Sourcecode-Debugger und Laufzeit-Optimierer integriert, dazu viele Fertig-Bibliotheken. Das Ganze sollte nach Möglichkeit modular oder besser noch objektorientiert arbeiten.
Und da kommt so ein respektloser CPC-Heini daher, schaltet sein eigentlich längst ausgestorbenes Rechner-Fossil ein und tippt in den völlig unzumutbaren Editor des eingebauten Basic-Interpreters ganz mühelos lauffähige Programme ein! Und, was das Härteste ist: Es scheint ihm auch noch Spaß zu machen...
Nichts gegen Profi-Systeme, und auch nichts gegen kommerzielle Software-Entwicklung mit Fertig-Bausteinen. Alles hat seinen Platz – und wer heute für den PC-Markt Programme entwickelt, hat ein hartes Brot. Aber: Wenn ich etwas ausprobieren oder irgendeine Bierlaune auf dem Computer umsetzen will, pfeife ich auf Performance und Stand der Technik und greife viel lieber zu meinem 8-Bit-Rechner mit seinem herrlich unmodernen eingebauten BASIC.
Wenn auch Sie sich nicht schämen, zu den lächelnden Dinosauriern zu gehören, sind Sie in diesem Heft goldrichtig: Da zieht sich das Stichwort "BASIC-Programmierung" wie ein roter Faden hindurch. Unser ***BASIC-Kurs*** *verpaßt Ihnen Tips, wie Sie Struktur und Wartungsfreundlichkeit in Ihre Programme bekommen. Die große* ***RSX-Bibliothek*** *zum Abtippen verhilft Ihrem CPC-BASIC zu all den Befehlen, die Sie sich schon immer gewünscht haben.*
Unsere kleinen Rechner haben immer noch ihr Recht. Sie erhalten uns den Spaß am "einfachen Computern" ohne Netz und doppelten Boden. Wenn es sie wirklich eines Tages nicht mehr geben sollte, wird die Computerwelt ein ganzes Stück ärmer sein.

Das meint jedenfalls Ihr

Peter Schmitz

Peter Schmitz, Chefredakteur

INHALT

BERICHT

"Englisches Wetter" für die CPCs 19
– Die "Business Computing 91" in London und Amstrads neue Heimcomputer-Strategie

Was macht der Rest der Welt? 112
– Eindrücke von der "Systems" in München

CPC-PROGRAMME:

20 **RSX-Professional**
– Die große Befehlsbibliothek für alle CPCs: Speicher und Bildschirm gekonnt manipuliert

30 **Elektronikers Freund**
– Widerstände verwalten, Farbcodes entschlüsseln, Kombinationen optimieren: die Alltagshilfe für den Bastler

34 **Frisch kopiert und editiert ...**
– Zwei nützliche Bonusprogramme auf der neuen DATABOX

CP/M:

Pascal-Kolleg: Schöne Schleifchen 35
– Schleifenkonstruktionen in Turbo Pascal und Basic

Alles in Ordnung? 36
– Sortieralgorithmen als Kraftfutter für Ihre selbstgeschriebene Datenbank

Viermal C 40
– Vier Compiler im Test: Small-C in kommerzieller und PD-Version, außerdem Arnor-C und Hisoft-C

Inline für Z80 44
– Assemblerprogramme in Turbo-Pascal-Code einbinden

SOFTWARE-REVIEW:

10 **Heißer Stoff für kalte Nächte**
– Passend zur Jahreszeit: abendfüllende Computerspiele für die CPCs

17 **Neues aus Frankreich**
– Ein Spiele-Hersteller aus dem Land der Gallier überraschte uns mit einigen brandheißen Ideen

KNOW-HOW:

Weg mit dem GOTO 72
– Basic-Kurs: So bringen Sie Struktur in Ihre Programme

Schwertgeschärfte Ohren 76
– Der Weg zum eigenen Rollenspiel-System: Texteingaben

Binärdateien trickreich verschlüsselt 80
– Programme schützen mit der "Alkatrax"-Codierung

Assembler-Ecke: Gewitzt und fix mit RSX 83
– Wie man dem CPC-BASIC mit einer simplen Maschinenroutine selbstgebastelte Befehle unterjubelt

Während man in Deutschland die 8-Bit-Computersysteme totschweigt, sah die Lage in England bislang noch besser aus. In London wurde jetzt von Amstrad eher lustlos der neue PCW präsentiert – wie sind die Aussichten für die 8-Bit-Freunde?
S. 19



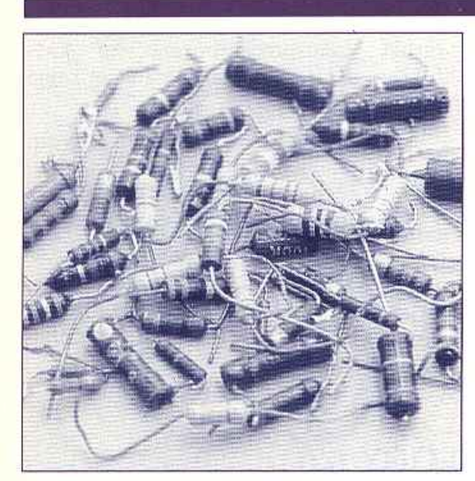


Eine der größten Stärken des im CPC eingebauten "Locomotive BASIC" ist seine Offenheit gegenüber Befehlserweiterungen. Unsere große Bibliothek "RSX-Professional" bringt Pulldown-Menüs, RAM-Disk-Funktionen, Speichermanipulation und noch viel mehr per BASIC-Befehl auf Ihren Rechner
S. 20

Ob Sie sich wie unsere "Kriegerin" im Bild aufmachen, um Monster und Bösewichte mit dem Schwert in der eigenen Hand zu bekämpfen – oder ob Sie sich vor Ihren CPC setzen, um programmierte Heldinnen und Helden in Ihrem ersten selbstgeschriebenen Rollenspiel zum Leben zu erwecken, das müssen wir wohl Ihnen überlassen. Auf jeden Fall finden Sie in der zweiten Folge unseres Rollenspiel-Kurses jede Menge Material für "Abenteuer satt". Diesmal geht es um die Analyse von Texteingaben
S. 76



Dezember '91/Januar '92



"Widerstand ist zwecklos" – diesen Satz könnte wohl kein Elektronik-Bastler unterschreiben. Ein ordentlicher Bestand an Widerstandswerten ist der Stolz jedes Lötkolbenschwingers. "Elektronikers Freund" verwaltet den Bestand und errechnet für fehlende Werte die günstigsten Ausweich-Kombinationen.

S. 30

Möchten Sie Ihre Programme vor den neugierigen Disassemblern böswilliger Mitmenschen schützen? Ein wirkungsvoller Kopierschutz setzt voraus, daß ein entsprechendes Disketten-Startprogramm sich in puncto Spurzugriff nicht in die Karten schauen läßt. Mit dem "Alkatrax"-Verfahren stellen wir Ihnen eine besonders ausgefuchste Methode zur Verschlüsselung von Binärdateien vor

S. 80

Gerade bei anspruchsvolleren Spielprogrammen passiert es auch dem ausgeschlafensten Computerfreund, daß er irgendwann nicht mehr weiter weiß. Für Spieletips und Lösungen gibt es bei uns die "Gamer's Message" – eine Rubrik zum Mitmachen!

S. 51



Service

AMS-Line 47
– Der "heiße Draht" zu den Spezialisten

PCW:

90 **Im Herzen des Joyce: Wie geht's dir, Rechner?**
– Den Status-Meldungen auf den Grund gegangen

96 **Ein Hauch von Lexus**
– Das programmierte Lexikon in der ultimativen Komfort-Version

102 **Der PCW als Springinsfeld**
– "Bounce" lädt Sie zu einem flotten Spielchen ein

107 **Keine Angst vor Plotter und Vektoren!**
– Ein komplettes Grafiksystem zum Abtippen, flexibel und leistungsfähig

114 **Für Sie gespielt**
– Die Gesellschaftsspiel-Klassiker "Trivial Pursuit" und "Scrabble" auf dem PCW

116 **LocoScript, zum 2.3ten**
– Ein Blick auf die neue Version der Standard-Textverarbeitung

TIPS & TRICKS:

100 DM für 1 kByte 48
- Kreisschrift
- Zahlenraten
- Magic Rings
- Schiffe versenken
- Mini-Vokabelpauker
- Bildschirmwechsel

Gamer's Message 51
– Die "Erste Hilfe" für den Spielefreund

Der Grafik-Schieber 70
– "Gigacad"-Grafikkonvertierung unter CP/M

RUBRIKEN:

3 **Editorial**
6 **Aktuelles**
71 **Angelesen**
86 **Jahresinhaltsverzeichnis: Das war 1991**
88 **Leserbriefe**
88 **Hotline**
95 **Händlerverzeichnis**
95 **Kleinanzeigen**
117 **Impressum**
118 **Inserentenverzeichnis**
118 **Vorschau**

Prowort 5.09

Die englische Firma Arnor, die bereits für den CPC als auch für den PCW das Textverarbeitungsprogramm Prowort entwickelte, hat nun die neueste Version für den PC fertiggestellt.

Dieses voll in Deutsch gehaltene Programm bietet unter anderem ein Lexikon mit 150.000 Wörtern, eine automatische Datensicherung, Fuß- und Endnotenverwaltung sowie eine umfangreiche Programmiersprache für die Serienbrieferstellung.

Weiterhin interessant sind Konvertierungsoptionen, die es möglich machen, die Texte, welche auf anderen Rechnern mit Prowort verfaßt wurden, in das MS-DOS-Format zu übertragen. Die ersten Erfahrungen in der Redaktion waren bei der Arbeit mit Prowort sehr positiv. Gerade für Vielschreiber sollte auch die Rechtschreibprüfung interessant sein, die durch unzählige Funktionen sicher an jeden Geschmack anpaßbar ist.

Hier sei zum Beispiel die On-Line-Prüfung genannt, die direkt beim Eingeben von unbekannten Wörtern in den Text durch einen kurzen Piepston aufmerksam macht. Nun kann man nachschauen und, wenn das Wort in Ordnung ist, einfach weiterschreiben. Man wird also beim Verfassen eines Dokuments nicht durch unnötige Bildschirmwarnungen, die eventuell sogar noch bestätigt werden müssen, gestört.

Info: Prowort Service, Postfach 1316, 8045 Ismaning, Tel. 089/966270



Spiele-PC von Amstrad

In England stellte die Firma Amstrad jetzt das PC5286 Games Pack vor. Hierbei handelt es sich um einen 286er PC mit 16 MHz und VGA-Grafik, die bei einer maximalen Auflösung von 800600 Punkten ganze 256 Farben darstellen kann. Weiterhin sind eine 40-MByte-Festplatte sowie ein 1,4-MByte-Laufwerk vorhanden. Im System integriert sind ebenfalls eine Ad Lib-kompatible Soundkarte mit zwei Lautsprecherboxen, ein AMSTRAD-Joystick sowie eine PS/2-Mouse.

Damit man auch gleich ein Spielchen wagen kann, finden sich noch die Spiele Links, F-15 Strike Eagle II sowie Price of Persia.

Ob dieses Gerät auch in deutschen Landen vertrieben wird, stand zum Redaktionsschluß noch nicht fest.

DISCOLOGY 6.0

Das aus der französischen Softwareschmiede Meridian stammende Disketten- und Dateikopierprogramm Discology 6.0 ist nun auch in französisch in Deutschland erhältlich. Damit der Anwender sich nicht durch die französischen Menüpunke quälen muß, erfolgt die Auslieferung des Programms mit deutschem Handbuch. Der Preis beträgt 99,90 DM für die Diskette, 25,- DM für ein Update von der Version 5.1 sowie 35,- DM für das Update von einer älteren Version.

Bezugsquelle: MB-Versand, Mike Behrend, Postfach 50 11 32, 5000 Köln 50

Das erste Stromautomobil

Wer meint, Tüftler und liebenswerte Gehirn-Akrobaten gäbe es erst seit der Einführung der Heimcomputer, muß sich beim Besuch eines jeden Technik-Museums immer wieder eines Besseren belehren lassen.

Ein Beispiel: Elektroantrieb wird von heutigen Autobauern als *die* Herausforderung der neunziger Jahre beschrieben. Eric Rainsford (75) kann da nur mit dem Kopf schütteln: Sein abgebildeter "HAWA" wurde von der Hannomag in Hannover in den Jahren 1921 bis 1924 gebaut. Dieser kleine Zweisitzer zeigt, daß auch lange vor Erfindung des "künstlichen Dummkopfs" Computer an ungewöhnlichen Ideen rund um fließenden Strom und fleißige Elektronen gefeilt wurde.

Das Fahrzeug wurde jetzt anläßlich der Jubiläumsfeier zum 750jährigen Bestehen der Stadt Hannover in einer Verkehrsausstellung gezeigt.

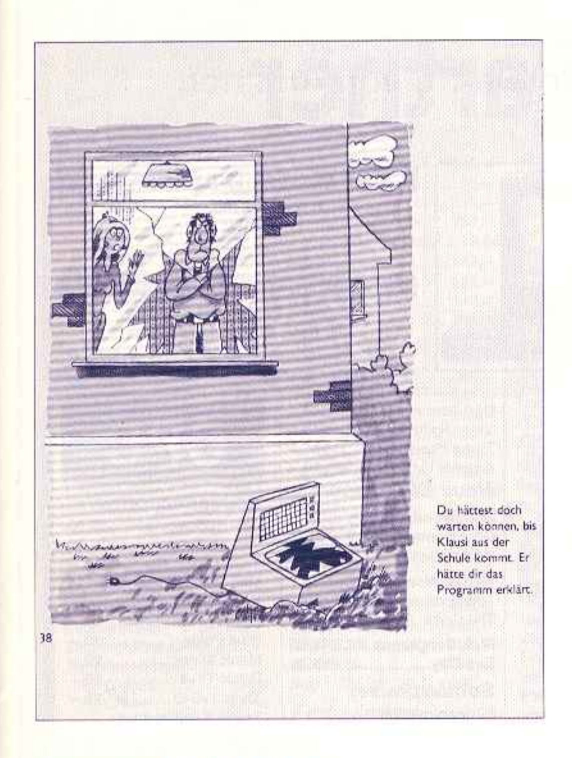
Neuer Druckertreiber für LocoScript

Für den Standard-Matrixdrucker des PCW 8.xxx gibt es jetzt – zusätzlich zu den übrigen Hervorhebungsmöglichkeiten wie Fettdruck, Unterstreichen, Breitschrift und Kursivdruck – unter LocoScript 2.28 sowie 2.30 die Möglichkeit, alle Zeichen auch doppelt hoch auszudrucken.

Möglich wird dies durch einen Ersatztreiber, der statt der Original MATRIX.PRI auf die LocoScript-Startdiskette kopiert wird. Der neue Treiber ist zum alten in allen NLQ-Druckeigenschaften identisch. Dies bedeutet, daß alle Texte für den einen Druckertreiber auch mit dem anderen ausgedruckt werden können. Auch Zusatzprodukte wie verschiedene Fonts werden wie gewohnt verwendet, und der Treiber ist kompatibel zum automatischen Einzelblatteinzug (vertrieben von JPS Bernhard Graßhoff).

Preis: zirka 78 DM
Bezugsquelle: Reiner Seitz, Software - Erstellung - Pflege - Schulung, Schongauer Str. 5, 8500 Nürnberg





COM'92

Vom 5.5 bis zum 9.5.1992 wird in Erfurt zum zweiten Mal die COM'92 stattfinden.

Die Messe umfaßt rund 20.000 qm Ausstellungsfläche, wo sicherlich auch dieses Jahr wieder jede Menge Interessantes zu lesen beziehungsweise auch zu hören sein wird. Im letzten Jahr waren zum Beispiel auch für CPC- und PCW-Besitzer einige interessante Firmen anwesend. Wer sich mehr in Richtung Basteleien ein Bild von der Messe machen wollte, konnte zum Beispiel den ROBOTRON-Stand besuchen. Also nichts wie hin.

LocoScript- "Bundles"

Nachdem die Preise für VORTEX-System-2000-Festplatten von der Firma Wiedmann, Ismaning, schon rapide herabgesetzt wurden, passiert jetzt auch ähnliches mit LocoScript.

Die Textverarbeitung in der Version 2.30 kann in sogenannten Bundles bezogen werden. LocoScript wird also in Verbindung mit einem der aus der Loco-Familie stammenden Programme zusammen günstiger verkauft.

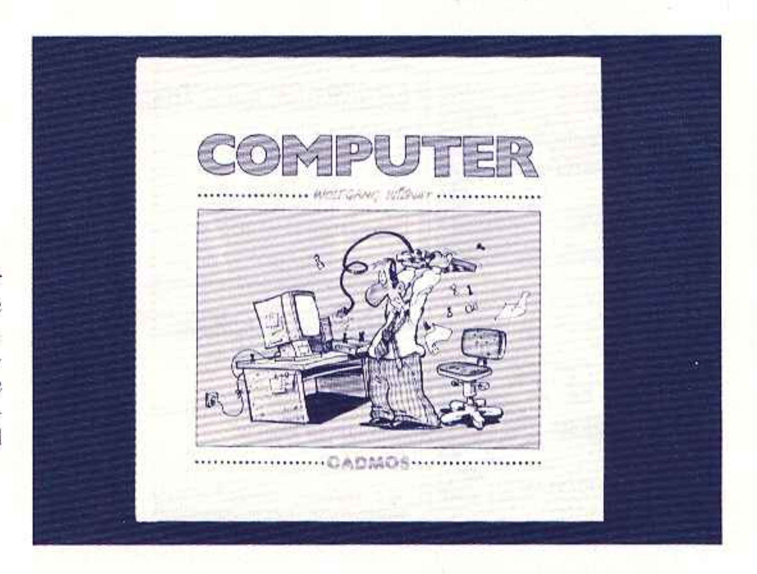
Die Verbindung LocoScript mit LocoFile oder aber LocoMail kostet jetzt nur noch 256,50 DM (316,20 DM vorher).

Bezugsquelle: Wiedmann Unternehmensberatung und Handel, Korbiniansplatz 2, 8045 Ismaning



Computer

Im Cadmos Verlag ist jetzt wieder ein Buch mit neuen Computercomics erschienen. Nachdem Titel wie "LIEBE ein Buch mit sieben Siegeln" oder "DIE PARTY geht jetzt richtig los" erschienen sind, widmet Wolfgang Willnat in COMPUTER seine ganze Aufmerksamkeit diesem Medium. Die 42 farbigen Illustrationen bringen sicherlich jeden Computerbesitzer zum Schmunzeln, aber auch zum Nachdenken. Freundlicherweise wurde uns vom Cadmos-Verlag die Genehmigung erteilt, eine der Grafiken in der CPC International abzudrucken, damit Sie sich selber ein Bild von diesem wirklich gelungenen Sammelwerk machen können. Sollte Ihnen jetzt noch ein Weihnachtsgeschenk für Ihre Liebsten fehlen, können wir dieses Buch nur empfehlen.



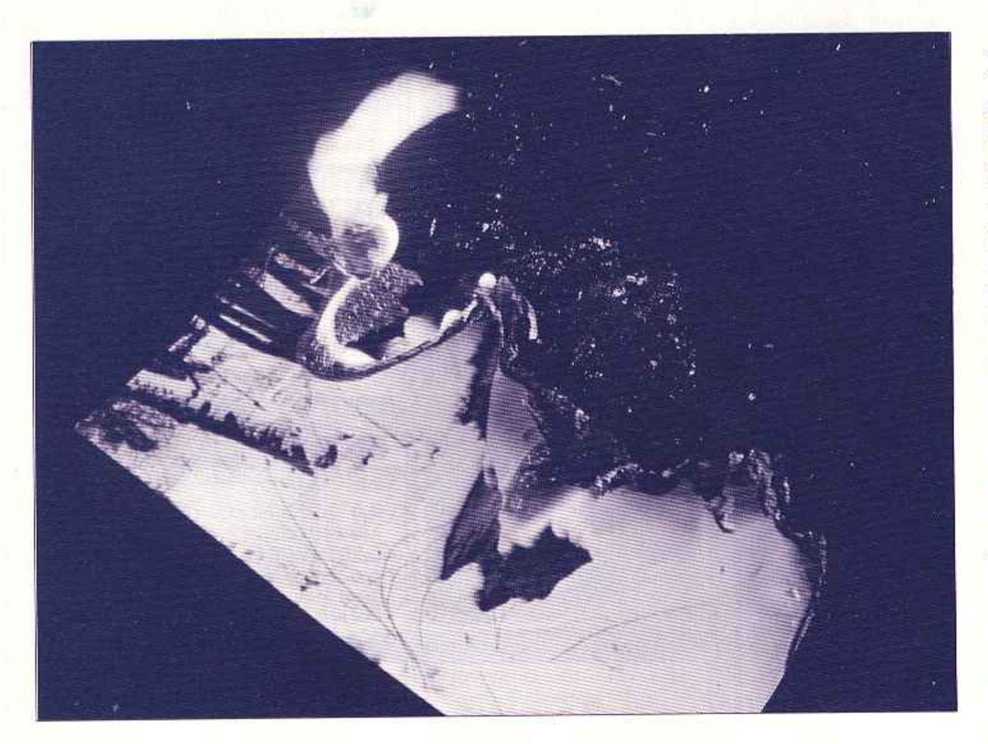
Neues Patent hilft Plastik sparen

Nach Aussagen der Firma Total Printstrument wiegt eine gewöhnliche Farbbandkassette für Drucker oder Schreibmaschine rund 150 Gramm. Erschreckend ist jedoch, daß sie ganze drei Gramm Farbe beinhaltet. Wird nun die verbrauchte Farbbandkassette gewechselt, landen ungewollt ganze 147 Gramm an Plastik und Farbband auf dem Müll. In den meisten Fällen wäre es jedoch ohne Probleme möglich, das Farbband und die Kassette noch ein- oder mehrmals zu benutzen. Was also fehlt, ist eine Möglichkeit, das Farbband wieder aufzufrischen.

Hier gibt es nun zwei Möglichkeiten. Bisher war es bei manchen Farbbändern möglich, diese einzuschicken und gegen entsprechendes Bares aufgefrischt zurückzubekommen. Dies ist umständlich und setzt voraus, daß noch ein zweites Farbband vorhanden ist.

Die Firma Total Prinstrument hat nun ein Patent angemeldet, das hier Abhilfe schaffen soll. So werden die Farbbänder von vorneherein mit einer Auffrischungstechnik versehen. Ist nun die Farbe aufgebraucht, entnimmt man der Kassette lediglich einen kleinen Auffrischungstank und tauscht diesen gegen einen neuen aus. Die Plastikverschwendung kann so wieder etwas verringert werden.

Weitere Informationen:
Total Prinstrument MFY Ltd.
Mr. T. Chow
Suite 7, City Garden Shopping Centre, 1/F.
233 Electric Road, North Point
Hong Kong



Heißer Stoff für kalte Nächte

Für Weihnachten oder einfach so: Unsere Spiele-Reviews zeigen, wo's lang geht!

Damit nicht auch die Gesichter lang werden, wenn die Nächte länger werden, haben wir uns wieder die Neuerscheinungen auf dem Spiele-Markt angesehen.

Angenehm aufgefallen sind dieses Mal die Anleitungen: Abgesehen von NORTH & SOUTH und BUILDERLAND liegen allen Spielen deutschsprachige Anweisungen bei.

Der Film **BATMAN** war trotz des riesigen Werberummels in Deutschland ein Flop. Trotzdem legt Ocean nun das gleichnamige Computerspiel vor.

Wenn es Nacht wird in Gotham City ...

Jack Napier war schon immer gemein und sadistisch. Doch seit seinem Unfall verkörpert er als *Joker* das Böse schlechthin. Um *Gotham City* zu retten, muß schnell gehandelt werden. *Bruce Wayne*, ein eher schüchterner Milliardär, weiß dies. Deshalb streift er nach Einbruch der Dämmerung sein Kostüm über und verwandelt sich in *Batman*. Doch außer einigem technischen Schnickschnack benötigt er viel Glück und Geschicklichkeit, um ans Ziel zu kommen.

Die fünf verschiedenen Levels geben fünf Situationen des Films wieder. Zu Beginn muß verhindert werden, daß *Jack Napier* eine Chemiefabrik überfällt. *Batman* kann dies zwar vereiteln, doch leider verschuldet er dabei selbst den Unfall, durch den *Jack Napier* in *Joker* verwandelt wird. Hierbei handelt es sich um ein Plattformspiel, bei dem es über Leitern und vorbei an zahlreichen Gegnern geht. Eine Besonderheit ist jedoch das *Batseil*, mit dem die Spielfigur hin- und herschwingen und sich auf höher gelegene Plattformen hieven kann.

Vick Vale, die hübsche Blondine, wurde inzwischen von *Batman* aus den Klauen des *Jokers* befreit. Nun muß dieser – das ist der zweite Teil des Spiels – mit seinem Spezialautomobil entkommen. Erstaunlicherweise handelt es sich in diesem Part nicht um ein Autorennen in der üblichen Perspektive. Vielmehr sieht man die Straße von schräg oben, was ein völlig anderes Spielgefühl vermittelt. Auch daß man abbiegen muß – und dabei Seile und Enterhaken zur Hilfe nehmen kann – setzt ungewöhnliche Akzente.

Nun hat *Joker* diverse Gegenstände mit einem Spezialgas vergiftet. Grund genug, schnellstens herauszufinden, um welche Gegenstände es sich dabei handelt. In diesem dritten Abschnitt muß gegen die Zeit geknobelt werden. Durch Ausprobieren aller Kombinationen kommt man aber ziemlich schnell ans Ziel.

Weiter geht's wieder auf den Straßen. *Joker* hat die Bevölkerung mit Versprechen ins Freie gelockt. Doch was in seine zahlreichen Luftballons gefüllt wurde, ist keine Luft, sondern Giftgas. Um die Gefahr für die Bürger zu bannen, muß *Batman* mit seinem Flugzeug, dem *Batwing*, die Halteseile der Ballons durchtrennen. Dabei muß er sich jedoch auch vor den Hubschraubern des *Jokers* in acht nehmen. Ähnlich wie der zweite Teil, aber die Ballons stellen eine neue Herausforderung dar.

Schließlich flüchtet *Joker* in die Kathedrale, wo es auf dem Dach zum großen Showdown zwischen ihm und *Batman* kommt. Wie zu Beginn muß sich *Batman* hier wieder mit seinem *Batseil* von Plattform zu Plattform schwingen, aber neue Gefahren sind hinzugekommen.

Ein altes Rezept – verschiedene Szenen eines Films durch unterschiedliche Spieletypen darzustellen – wurde bei BATMAN sehr gekonnt angerichtet. Jede der einzelnen Sequenzen bietet gegenüber anderen Programmen einige Eigenarten. Jeder Teil wurde mit einem eigenen Musikstückchen bedacht. Die Grafik ist farbenfroh und abwechslungsreich, wenn auch teilweise etwas grob. Kein besonderes Wunderwerk ist das Scrolling, es stört aber auch nicht gerade.

Spaß und Spannung hoch fünf

Nun könnte man BATMAN beinahe uneingeschränkt empfehlen. Das heißt, wer auf die Disketten-Version zurückgreifen kann, bekommt für sein Geld ein wirklich gutes Spiel. Auf Cartridge sind jedoch keinerlei Änderungen festzustellen – mit Ausnahme der typischen Cartridge-Verpackung und der ebenso dafür typischen Anleitung. Cassetten- und Disketten-Benutzer müssen sich allerdings darauf einstel-

len, daß die Levels nachgeladen werden.

BATMAN

Hersteller: Ocean-Software
Steuerung: Joypad, Joystick oder Tastatur
Voraussetzung: für alle CPCs, GX 4000
Preis: 98,00 DM (Cartridge)
Bezugsquelle: EDV-Obermeier
Bemerkung: Cartridge identisch mit der preisgünstigeren Disketten-Version
Bewertung:
Grafik: 3 Sound: 2
Präsentation: 2 Motivation: 2
Endnote: 3

Straßensperren sind kein Hindernis in **CRAZY CARS II**: Auf der Jagd nach Autoschiebern geht es durch vier Staaten der USA. Zu allem Unglück sind diese Autoschieber aber Polizisten, und so sind Sie ganz auf sich allein gestellt – und auf Ihren *Ferrari F40*.

Allein gegen die Polizei!

Tatsächlich liegt CRAZY CARS II das Straßennetz der Vereinigten Staaten zugrunde. Start- und Zielorte werden vom Programm angezeigt. Anhand der jederzeit abrufbaren Straßenkarte muß sich der Spieler nun selbst orientieren. An Ausfahrten wird die Nummer der abzweigenden Straße angezeigt.
Die Fahrt ist natürlich ein Kampf gegen die Zeit. Jede Explosion des Autos infolge eines Zusammenpralls mit Sperren oder anderen Wagen kostet wertvolle Sekunden, bis der Ferrari wieder einsatzbereit ist. Wenn das nur in Wirklichkeit auch so einfach wäre!

Übung macht den Meister

Zu einem schnelleren Spielende kann es kommen, wenn nach einem Unfall die Polizei zur Stelle ist und Sie verhaftet. Dann geht es wieder ganz von vorne los! Anfangs stellt sich oft das Gefühl ein, den Hindernissen ausgeliefert zu sein und auch mit Geschicklichkeit nicht viel ausrichten zu können.
Mit etwas Übung klappt es aber doch. Dann dauern auch die Spiele recht lange, weil die zurückzulegenden Distanzen groß sind.
Der 3D-Effekt beim Fahren ist gut gelungen. Hintergrundbilder und Randmarkierungen wechseln, wenn man in einen neuen Staat kommt.
Während in *Colorado* die Straße von Bäumen gesäumt ist, finden sich in *New Mexico* Kakteen. Auch wird man für die Möglichkeit, die Musik zugunsten von Sound-Effekten abzustellen, dankbar sein. Überraschend ist hier vor allem, wie gut das Näherkommen der Polizeifahrzeuge durch lauter werdende Sirenen angezeigt wird. Anzeichen für die verbesserten Hardware-Eigenschaften kann man jedoch lange suchen.
Bleibt die Frage, ob sich CRAZY CARS II am Markt behaupten kann. Besitzer der Console und der Plus-Geräte bekommen beim Kauf ein Spiel, das CRAZY CARS II ebenbürtig ist. So ist es wohl eher ein Fall für Leute, die auf Rennspiele versessen sind und sich an BURNIN' RUBBER sattgespielt haben.

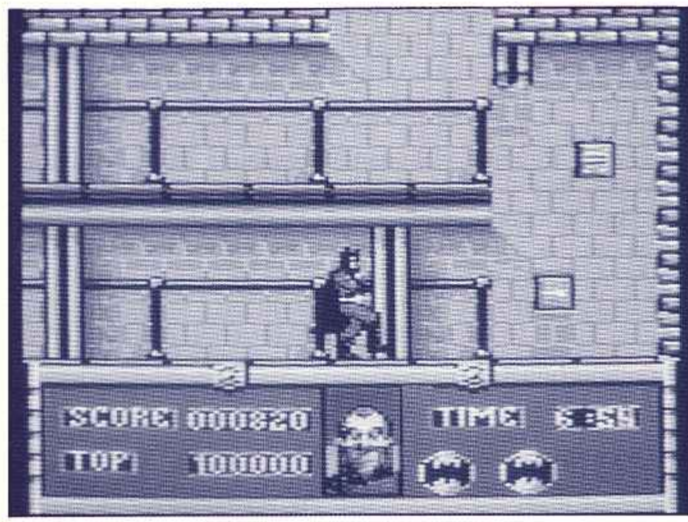
CRAZY CARS II

Hersteller: Titus
Steuerung: Joypad
Voraussetzung: GX 4000, 464 Plus, 6128 Plus
Preis: 98,00 DM (Cartridge)
Bezugsquelle: EDV-Obermeier
Bewertung:
Grafik: 2 Sound:3
Präsentation: 2 Motivation: 3
Endnote: 3

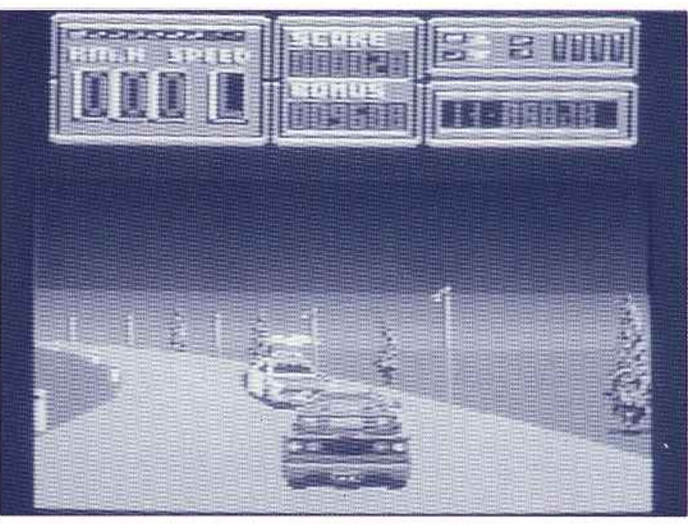
Schon wieder ein Spiel zum Film: **DICK TRACY**. In der Rolle des berühmten Detektivs müssen Sie ergründen, was es mit dem Verschwinden *Lips Manils* auf sich hat. *Big Boy Caprice* verkündet, daß die gegen ihn geäußerten Verdächtigungen haltlos seien – und heckt neue Pläne aus. Auch die anderen Personen werden in der Anleitung beschrieben, die bezaubernde *Tess Trueheart* ebenso wie der smarte Gauner *The Brown*. Das Szenario wird derart gesetzt, daß man fast schon mit einem Adventure-ähnlichen Spiel rechnen könnte. Doch weit gefehlt!

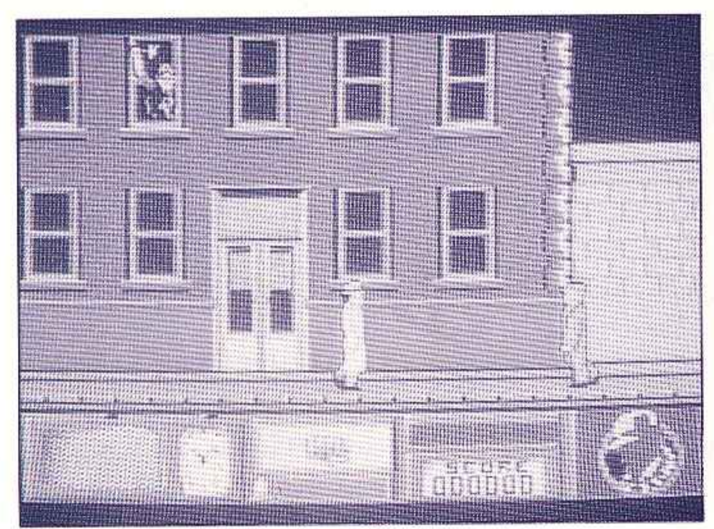
Alles schon mal dagewesen

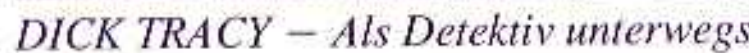
Der Spielablauf von DICK TRACY ist altbekannt. Anfangs unbewaffnet ziehen Sie durch die Straßen und Cafés, müssen die Ganoven mit der bloßen Faust von sich halten. Hin und wieder ergibt sich die Gelegenheit, Waffen aufzusammeln. Das Spielfeld scrollt horizontal, und Gegner gibt es genug. Wenn die Figur von einem Schuß getroffen wird oder einen Feind berührt, verliert sie Energie. Wenn diese aufgebraucht ist, endet das Spiel. Daß der Spieler nur ein Leben hat, ist manchmal etwas frustrierend. Vor allem weil es Stellen gibt, an denen man ohne Waffen gar nicht vorbeikommen kann. Und wenn man gerade vorher seinen letzten Schuß verbraucht hat, kann einen das ganz schön ärgern.
Weiterhin ist es nicht besonders schön, daß man es wesentlich leichter hat, wenn man mit ein paar Tricks spielt. Vor allem auch deshalb, weil durch diese Tricks der Spielfluß aufgehalten wird.
Die Schüsse und Schläge aus dem Laut-



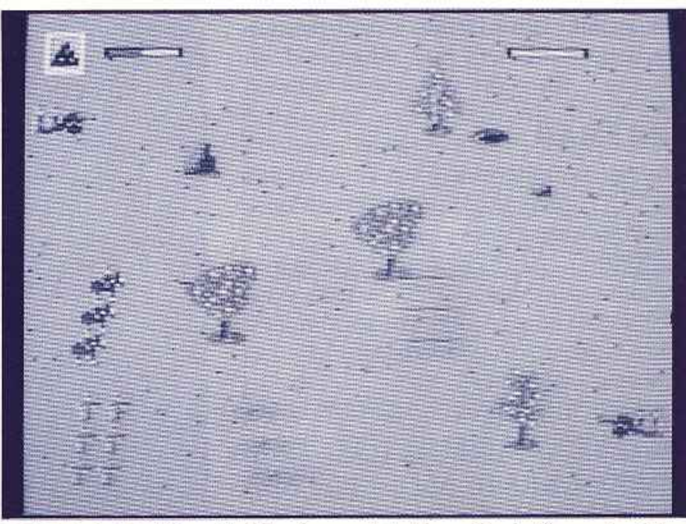
BATMAN – Eine fünfteilige Reise durch Gotham City


CRAZY CARS II – Verbrecherische Polizisten sind hinter Ihnen her!





DICK TRACY – Als Detektiv unterwegs



NORTH & SOUTH – Viel Spaß verspricht dieses Comic-Strategiespiel

sprecher sind zwar nicht gerade spektakulär, aber der Melodie auf jeden Fall vorzuziehen. Etwas aufgewertet wird das laue Spielprinzip dadurch, daß endlich die erweiterte Farbpalette, die bei Cartridge-Software zur Verfügung steht, ausgenutzt wird. Trotz hoher Auflösung sind Figuren, Häuser und Innenausstattung des Cafés farbenprächtig. Auch das Scrolling ist butterweich.

Seit Jahren kommen immer wieder ähnliche Spiele auf den Markt, die teilweise wesentlich abwechslungsreicher und spielerisch überzeugender sind. Da es aber auf Cartridge noch nicht so viele Spiele dieser Art gibt, ist DICK TRACY vor allem für Besitzer der GX-4000-Console geeignet, die ja nicht auf das große Angebot der Cassetten- und Disketten-Software zurückgreifen können.

DICK TRACY

Hersteller: Titus
Steuerung: Joypad
Voraussetzung: GX 4000, 464 Plus, 6128 Plus
Preis: 98,00 DM (Cartridge)
Bezugsquelle: EDV-Obermeier

Bewertung:
Grafik: 2 — Sound: 4
Präsentation: 2 — Motivation: 4
Endnote: 3

Erinnern Sie sich noch an "Fackeln im Sturm"? Etwas weniger kitschig, dafür lustiger geht es bei **NORTH & SOUTH** um das gleiche Thema, den Sezessionskrieg. Ganz im Stil eines französischen Comics ("Les Tuniques Bleues") ist dieses Strategiespiel mit Action-Einlagen gehalten. Das beginnt mit Verpackung und Anleitung und zieht sich durch das ganze Programm. Sofort nach dem Laden fällt auf, wieviel Mühe sich die Programmierer gegeben haben. Drei Sprachen (Englisch, Französisch und Spanisch) stehen zur Auswahl. Alle Tasten können selbst definiert werden, was bei Spielen für zwei Spieler leider immer noch nicht selbstverständlich ist. Statt eines menschlichen Gegenübers springt auch der Computer ein. Die Fähigkeiten der beiden Seiten können verändert werden, was den Schwierigkeitsgrad ebenso verändert wie die Wahl des Jahres, in dem das Spiel beginnen soll. Des weiteren kann eingestellt werden, ob bestimmte Ereignisse, beispielsweise Indianerangriffe, zufällig auftreten sollen oder nicht. Auch die Action-Sequenzen können auf Wunsch weggelassen werden.

Ziel jedes Spielers ist es, das Gebiet der Vereinigten Staaten zu besetzen beziehungsweise alle Armeen des Gegners zu besiegen. Nachschub gibt es über Bahnlinien, daher ist es wichtig, eine solche zu besitzen und zu sichern. Je mehr Länder man kontrolliert, desto schneller bekommt man auch neue Armeen. Treffen gegnerische Einheiten auf einem Land zusammen, blendet der Computer eine andere Darstellung ein. Das Schlachtfeld erscheint, und beide Spieler können ihre Kanonen, Reiter und Fußsoldaten so einsetzen, daß sie beim Gegner den größtmöglichen Schaden anrichten. Hier kommt es auf ein gutes Reaktionsvermögen an. Dies gilt auch für Zugüberfälle und Angriffe auf Forts.

Einiges an diesem Programm erinnert an den Brettspiel-Klassiker "Risiko", doch es ist beileibe keine billige Kopie oder Computer-Umsetzung. Freunde traditioneller Kriegsspiele kommen wahrscheinlich weniger auf ihre Kosten, denn das Spiel ist eher unterhaltend als komplex.

Beste Unterhaltung für viele Stunden

NORTH & SOUTH verbindet wie kein anderes Spiel auf dem CPC Strategie mit guter, humorvoller Grafik und Action. Alle, die bisher Strategiespiele langweilig fanden, werden eines Besseren belehrt werden. Man wird vor allem zu zweit viele spannende Stunden haben.

NORTH & SOUTH

Hersteller: Infogrames
Steuerung: Tastatur oder Joystick und Tastatur
Voraussetzung: 128 kByte
Preis: 84,50 DM
Bezugsquelle: EDV-Obermeier
Bemerkungen: nur auf Diskette erhältlich

Bewertung:
Grafik: 2 — Sound: 3
Präsentation: 1 — Motivation: 1
Endnote: 1

Wie anscheinend bei Umsetzungen von Spielautomaten üblich, glänzt auch **BLACK TIGER** nicht mit einer besonders originellen Rahmenhandlung. Böse Drachen überfielen die Erde, richteten Verwüstung an und stürzten die Welt in ein finsteres Zeitalter. Nach einiger Zeit findet sich ein Held, der bereit ist, das Land wieder etwas wohnlicher zu gestalten und die Störenfriede zu beseitigen. In diesem Fall heißt dieser Held *Black Tiger*.

Bekannte Story, bekanntes Spielprinzip

Doch bei einer Automaten-Umsetzung sollte man nicht so sehr auf die Story

achten. Gute Grafik, abwechslungsreiche Gegner und ein flotter Spielablauf machen einen gutes Arcade-Spiel aus. Im Idealfall steckt sogar noch eine originelle Spielidee dahinter.

Doch leider hat BLACK TIGER auch auf diesen Gebieten nichts vorzuweisen. Zwar hat es einige wenige Eigenheiten – das Spielfeld scrollt auch nach oben und unten, wodurch das Spiel noch einige Plattform-Elemente enthält – und relativ viele unterschiedliche Feinde. Doch durch eine scrollende Landschaft laufen, Extrawaffen einsammeln oder einkaufen, Bonuspunkte ergattern und tote Feinde am Wegesrand lassen, gab es schon oft. Selten jedoch mit so einfarbiger Grafik – das Spielfeld ist grün und schwarz, ohne Schattierungen – und einem sehr ruckigen Scrolling. Wo andere Spiele mit realistisch dargestellten Animationen aufwarten, zeigt sich bei BLACK TIGER nur eine merkwürdige Bewegung der Beine. Die Steuerung ist recht ungenau. Sound-Effekte fehlen völlig. Auch die Musik tut nicht viel, um den miserablen Gesamteindruck aufzubessern. Einzeln wären diese Punkte vielleicht noch zu verkraften, aber leider werden sie durch keine Vorteile gegenüber anderen Action-Games wettgemacht.

Schwarzer Tiger oder lahme Ente?

Wie war das doch gleich auf der Verpackung? "U.S. Gold have created the Gold Standard. Others can only seek to emulate." Wenn sich andere Software-Häuser wirklich anstrengen müßten, um die Qualität von BLACK TIGER zu erreichen, würde ich noch heute meinen CPC verkaufen.

BLACK TIGER

Hersteller: US Gold
Steuerung: Joystick oder Tastatur
Voraussetzung: für alle CPCs
Preis: 61,50 DM
Bezugsquelle: EDV-Obermeier
Bemerkung: lädt nach

Bewertung:

Grafik: 4	Sound: 4
Präsentation: 4	Motivation: 3
Endnote: 4	

Auch in **SWITCHBLADE** geht es um eine altbekannte Story. Die sechzehn Teile eines Schwertes müssen gefunden werden, um *Havok* zu besiegen. Wer das ist, wird in der Anleitung nicht erwähnt. Aber er hat den Tod einiger Leute auf dem Gewissen und auch sonst einiges Unheil angerichtet. Nach seiner Rückkehr ist die zehntausendjährige Phase des Friedens vorbei. Das dürfte wohl Grund genug sein, ihn ein für allemal an seinem schändlichen Treiben zu hindern.

Im Kampf gegen den Tyrannen

Der größte Teil des Spiels findet unterirdisch statt. Dort gibt es ein weitverzweigtes System von Räumen und Gängen, in denen allerlei Gefahren lauern. Diese reichen von Flammen über Fantasy-Lebewesen bis hin zu Robotern. Die einzelnen Teile der Räume sind über Leitern verbunden, außerdem kann die Spielfigur springen.

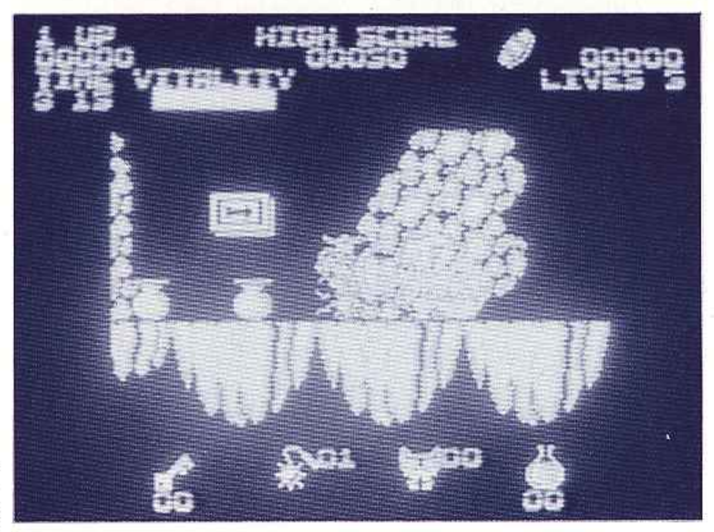
Unser Held kann sich anfangs mit Fausthieben und Fußtritten zur Wehr setzen. Später hat er die Gelegenheit, sich mit verschiedenen Schußwaffen effektiver zu verteidigen. Je länger der Feuerknopf gedrückt wird, desto kräftiger der Schuß oder Schlag. Auf diese Weise läßt sich auch die Tritthöhe regulieren. Andere Extras verstärken die eigene Panzerung oder steigern die Durchschlagskraft der Waffen. Durch Aufsammeln von Buchstaben lassen sich Bonuspunkte, aber auch Freileben erhaschen.

Doch viele dieser Hilfen fallen unserem Helden nicht in den Schoß. Vielmehr sind einige von ihnen unter Steinen versteckt, die zuerst zerschlagen werden müssen. Und wer einmal kräftig gegen die Wand tritt, findet bei dieser Gelegenheit vielleicht sogar einen Geheimgang. Eine weitere Besonderheit des Spiels ist, daß nur die Räume sichtbar sind, die man schon besucht hat. Wenn man also in einem kleinen Raum steht, der nur einen Bruchteil des Bildschirms füllt, sieht man anfangs nur die Ausgänge. Wenn man dann einen davon durchschreitet, wird schlagartig der neu betretene Raum sichtbar, wodurch man hin und wieder unangenehme Überraschungen erleben kann. Der alte Raum bleibt natürlich sichtbar.

Das Erforschen der Räume macht Spaß, besonders auch die Suche nach Geheimgängen. Der Schwerpunkt des Spiels liegt eindeutig auf Action, doch wer ein bißchen nachdenkt und mit der richtigen Taktik vorgeht, kommt an einigen Stellen viel leichter vorbei. Angenehm auch, daß man bei jedem Spiel etwas weiterkommt und trotzdem immer wieder neue Herausforderungen findet. Auch wenn das Spielprinzip altbekannt klingt, spielt sich SWITCHBLADE doch erfrischend anders als Konkurrenzprodukte. Erfreulich ist die Liebe zu Details bei der Präsentation, die vom Aufbau der Titelgrafik bis zur Einführung ins Spielgeschehen gelungen ist. Auch an eine Highscore-Liste wurde gedacht.

Ausgefeilt!

Besitzern von 128-kByte-Rechnern fällt die Wahl zwischen den knackigen Sounds und dem ebenso gelungenen Musikstück sicher schwer. Wer nur 64 kByte Speicher hat, muß sich mit Effekten begnügen. Ein besonderes Lob gilt aber dem Grafiker. Wie hier der vierfarbige Mode 1 des CPC ausgereizt wurde, setzt neue Maßstäbe. Geschickte Schattierungen lassen die Steine solide, ja beinahe plastisch wirken. Naturgemäß ist der Bildschirm zwar nicht knallbunt, dafür paßt aber die Farbwahl sehr gut zum Spiel. Und da einige zusätzliche Farben von Bild zu Bild variiert werden, ist zusätzlich für Abwechslung gesorgt. Man findet wenig, was noch zu verbessern wäre. Bravo!



BLACK TIGER – Eine lahme Automatenumsetzung

SWITCHBLADE

Hersteller: Gremlin
Steuerung: Joystick
Voraussetzung: für alle CPCs
Preis: 61,50 DM
Bezugsquelle: EDV-Obermeier
Bemerkung: auch – mit leicht verbesserter Grafik – auf Cartridge erhältlich

Bewertung:

Grafik: 1	Sound: 2
Präsentation: 1	Motivation: 1
Endnote: 1	

MOONWALKER heißt der erste Kinofilm des scheuen Superstars – oder muß man schon Megastar sagen? – *Michael Jackson*. Während andere Musiker versuchen, möglichst viele Fans zu haben, um sich in deren Anbetung sonnen zu können, ist *Michael Jacksons* größtes Problem, seinen Fans zu entkommen.

Mr. Big ist ein unfreundlicher Zeitgenosse: Er handelt mit Drogen. Außerdem, was noch schlimmer ist, verrät er *Michaels* Versteck dessen Fans.

Auf der Flucht vor Fans

Das Spiel beginnt in den Filmstudios, wo unter anderem Teile eines Kaninchenkostüms eingesammelt werden müssen, die *Michael* benötigt, damit ihn seine Anhänger nicht erkennen.

Mit einem Motorrad geht die Flucht aus der Stadt weiter. Dieses kann durch Aufsammeln von Kugeln in ein Auto verwandelt werden, das in der Lage ist, Straßensperren zu durchbrechen. Für zerstörte Drogenverstecke gibt's Bonuspunkte.

Im *Club 30* angekommen, kann sich unser Star nicht etwa ausruhen. Vielmehr muß er eine Bewaffnung finden, um es mit *Mr. Bigs* Bande aufnehmen zu können. Beim Finale in der Arena sieht sich *Michael Jackson* plötzlich einer Plasma-Kanone gegenüber. Diese soll zerstört werden, damit er sich dann auf die Auseinandersetzung mit *Mr. Bigs* Truppen konzentrieren kann.

Nach dem eindrucksvollen Vorspann folgt bald die Ernüchterung. Der erste Teil ist quälend langweilig. Das Labyrinth – von oben gesehen – ist riesig und die Spielfigur langsam, so daß dieser Teil endlose Längen hat. Die Feinde sind so dünn gesät, daß man manchmal eines der zwanzig (!) Leben verliert, weil man gar nicht mehr damit gerechnet hat, daß man auch noch auf Gegner achten muß.

Etwas besser geht es dann im zweiten Teil weiter, der dem ersten ähnlich, aber doch anspruchsvoller ist.

Nach dem schwachen Anfang wird's besser

Erst im dritten und vierten Teil kommt wirklich Action auf. Nur wegen diesen lohnt es sich auch, MOONWALKER zu spielen. Doch kann einem schnell die Lust vergehen, sich jedes Mal wieder eine Viertelstunde durch Level 1 zu quälen.

Wenn man diesen mit einem Code überspringen könnte, wäre viel gewonnen.

Beim CPC 464 funktioniert die Nachladeroutine nicht, während es auf dem CPC 6128 keine Probleme gibt. Die Grafik ist durchweg gehobener Standard.

Merkwürdig ist nur, daß in einem Spiel, in dem es um einen Musikstar geht, die Musik so schlecht ist. Sie steigert sich zwar wie das Spiel selbst gegen Ende, aber im ersten Teil wird man sie nach wenigen Sekunden abschalten.

MOONWALKER

Hersteller: US Gold
Steuerung: Joystick oder Tastatur
Voraussetzung: CPC 6128
Preis: 61,50 DM
Bezugsquelle: EDV-Obermeier
Bemerkung: lädt nach

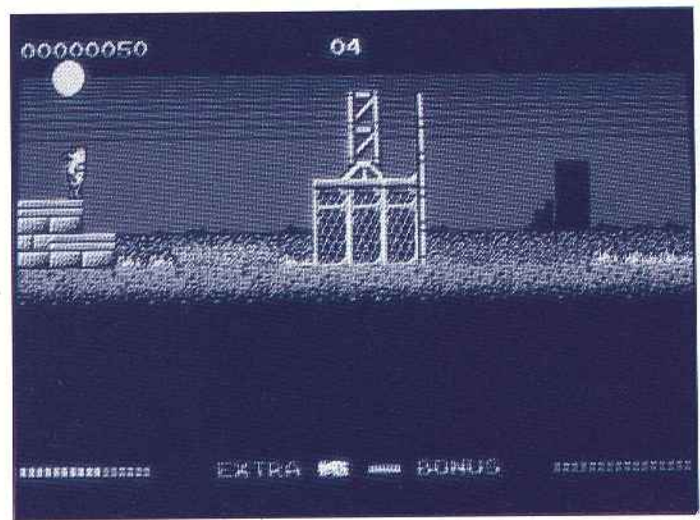
Bewertung:

Grafik: 3	Sound: 3
Präsentation: 3	Motivation: 3
Endnote: 3	

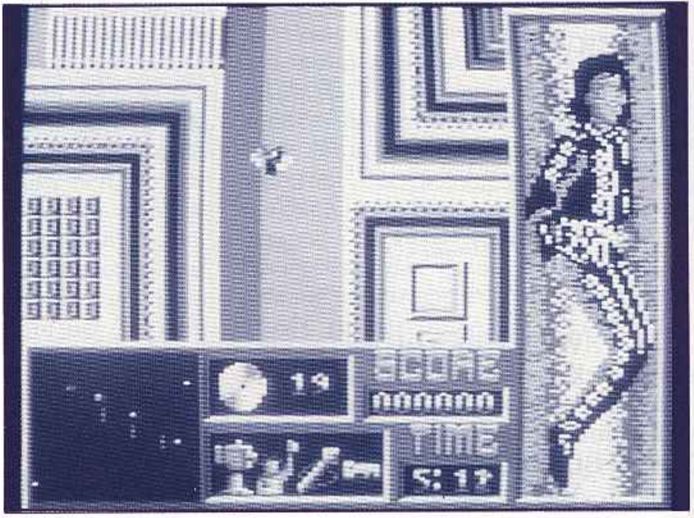
Überraschend ausführlich und stimmungsvoll ist die Einführung in die Handlung von **VENDETTA**. Sie sind ein ehemaliger Kriegsheld und nun ein Ausgestoßener aus der Gesellschaft, weil Ihre Erfolge die Ihrer Vorgesetzten überstiegen haben und diese Ihnen das Leben zur Hölle machen wollen. Ihr Bruder arbeitet als Professor für das Verteidigungsministerium an hochbrisanten Entwicklungen. Als er eines Tages von Terroristen entführt wird, ist für Sie die Stunde der Rache gekommen. An seinen Entführern. Und an Ihren ehemaligen Vorgesetzten.

Rache ist süß!

Von schräg oben gesehen ist VENDETTA ein dreidimensionales Arcade-Adventure. Während Sie Gebäude und Schubladen durchstöbern, müssen Sie sich vor den Terroristen in acht nehmen oder sich diesen tat- und schlagkräftig entgegenstellen. Doch bei allen Kämpfen sollte die eigentliche Mission nie vergessen werden, zumal die Zeit unaufhaltsam verstreicht. Um an manche Gegenstände zu gelangen, ist die Bedienung eines herumstehenden Computers erforderlich. Doch dieser beginnt nach dem Einlegen einer Diskette noch nicht zu arbeiten, son-



SWITCHBLADE – Langweilige Story, spannendes Spiel



MOONWALKER – Der scheue Michael Jackson ist auf der Flucht vor seinen Fans

dern erwartet zuvor die Eingabe einer Codezahl. Diese muß erst gefunden werden!

Nach einiger Zeit gelingt es vielleicht auch, den Rennwagen einsatzbereit zu machen. Dann können Sie in andere Teile der Stadt fahren und dort die Suche fortsetzen. Dazu schaltet der Computer in die übliche Darstellung eines Rennspiels um.

VENDETTA ist genauso knifflig und komplex, wie es klingt. Es hätte vom Spielaufbau her das Zeug zu einem Superspiel. Leider ist aber die Programmierung etwas daneben gegangen. Daß der Lautsprecher fast stumm bleibt und die Grafik weitestgehend schwarzweiß ist, könnte man bei dieser Art von Spiel vielleicht noch verkraften. Ein großes Manko ist jedoch die Kollisionsabfrage. Diese führt nicht zum Tod des Spielers. Aber manchmal kann man durch Gegenstände einfach durchlaufen, an anderen Stellen ist der Weg blockiert, obwohl der Bildschirm frei ist. Das kann die Suche manchmal ziemlich erschweren.

Wenn man sich eine Weile mit VENDETTA beschäftigt, gewöhnt man sich an diese Mängel ein wenig. Wer so viel Zeit aufbringt und nach der ersten Enttäuschung weiterspielt, bekommt hier aber ein Programm, das auch Tüftler längere Zeit beschäftigen dürfte.

Ebenfalls von System 3 kommt **TUSKER**, das in die gleiche Kerbe wie VENDETTA schlägt. Ort der Handlung ist diesmal Afrika.

Sein Leben lang war Ihr Vater ein Abenteurer. Zu seinem großen Leidwesen teilten Sie jedoch diese Leidenschaft nie. Doch von seiner letzten Reise ist Ihr Vater nie zurückgekehrt. Sie stöbern in seinen Unterlagen und finden sein Tagebuch, aus dem einige Seiten herausgerissen sind. Es gibt Anhaltspunkte dafür, daß Ihr Vater dicht daran war, den legendären Elefantenfriedhof zu finden. Da flammt sein Geist in Ihnen auf, und Sie machen sich auf den Weg nach Afrika, um seinen Spuren zu folgen.

Startschwierigkeiten durch fehlerhafte Anleitung

Das Szenario für das Spiel wird schön gesetzt, und einige Tips erleichtern den Einstieg in das Geschehen. Die Anleitung hat aber auch ein großes Manko. Laut dieser kann man mit den Tasten F1 und F7 die Objekte umschalten und mit der Leertaste herumliegende Gegenstände aufsammeln. Daß die Objekte in Wirklichkeit mit den Tasten 1 und 2 umgeschaltet werden, findet man durch Ausprobieren noch ziemlich schnell heraus. Bis man weiß, wie man etwas aufsammeln kann, verliert man jedoch einige Nerven. Tatsächlich geht es, indem man den Joystick schräg nach rechts unten oder links unten bewegt und gleichzeitig den Feuerknopf drückt.

Beim Spielen muß man sich vor Arabern, Wächtern und allerlei anderen Gefahren hüten. Auch auf den Wasservorrat sollte man ein wachsames Auge haben. Ans Ziel gelangt man jedoch nur, wenn man den richtigen Gegenstand am rechten Ort verwendet. Das stellt den Spieler ab und zu vor größere Probleme. Dieses Ausprobieren der Wirkung der Hilfsmittel und das Erkunden der Lokalitäten machen den Reiz von TUSKER aus.

Spaß für mehr als nur einen Nachmittag

Auch wenn an Farben ziemlich gespart wurde, sind grafisch gute Ansätze vorhanden. Andere Stellen sind jedoch etwas ungelenk gezeichnet. Musik spielt nur direkt nach dem Laden, und während des Spiels gibt der Computer nur selten Töne von sich. Trotzdem kommen Spieler, die gerne herumforschen und ausprobieren, bei TUSKER auf ihre Kosten.

Abschließend bietet sich noch ein Vergleich mit VENDETTA an, das vom Spielprinzip her sehr ähnlich angelegt ist. Bei TUSKER fällt der Einstieg wesentlich leichter, was auch an der um Längen besseren Programmierung liegt. Dafür ist VENDETTA wesentlich komplexer. Welchem Pogramm man den Vorzug gibt, hängt nicht zuletzt davon ab, worauf man persönlich den größeren Wert legt.

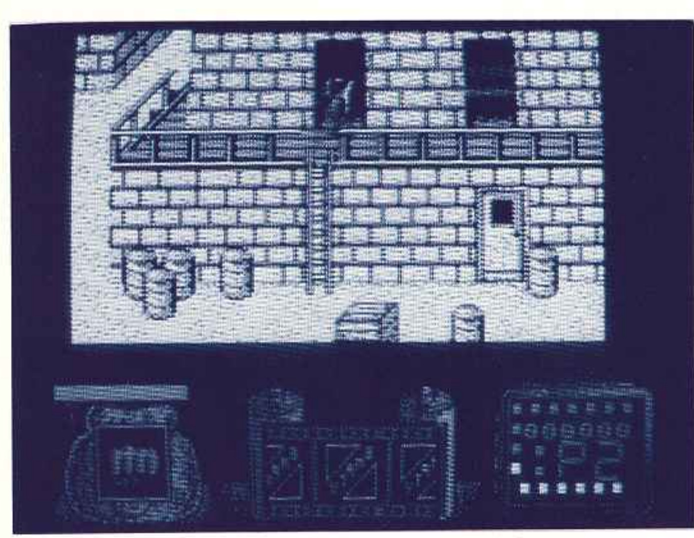
VENDETTA

Hersteller: System 3
Steuerung: Tastatur oder Joystick und Tastatur
Voraussetzung: für alle CPCs
Preis: 69,95 DM
Bezugsquelle: EDV-Obermeier
Bemerkung: lädt nach
Bewertung:
Grafik: 4 — Sound: 5
Präsentation: 3 — Motivation: 2
Endnote: 3

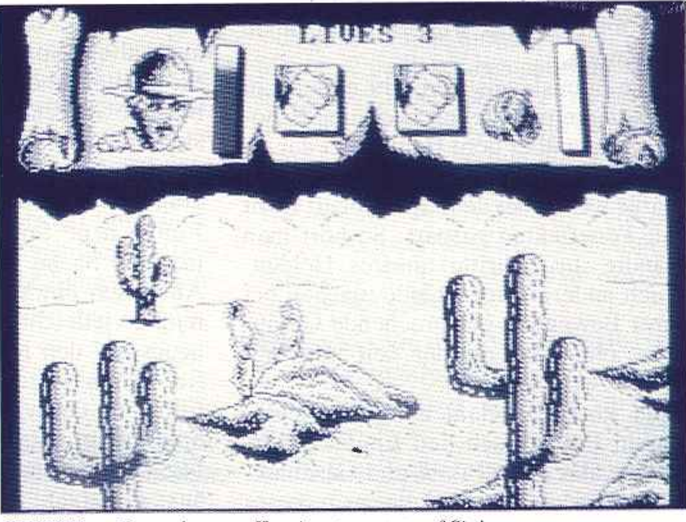
TUSKER

Hersteller: System 3
Steuerung: Joystick oder Tastatur
Voraussetzung: für alle CPCs
Preis: 61,40 DM
Bezugsquelle: EDV-Obermeier
Bemerkung: lädt nach
Bewertung:
Grafik: 3 — Sound: 4
Präsentation: 3 — Motivation: 2
Endnote: 3

Aus Frankreich bekamen wir auch etwas Neues. Das Spiel **BUILDERLAND** ist die Antwort auf das beliebte



VENDETTA – Retten Sie Ihren Bruder. Und Ihren Ruf.



TUSKER – Der schwarze Kontinent wartet auf Sie!

Spiel Lemmings. Die Aufgaben liegen nur etwas anders. Bei Lemmings müssen Sie dafür sorgen, daß die Lemminge sich selbst den Weg bahnen. Bei BUILDERLAND müssen Sie dem Helden des Spiels mit Hilfe des Joystiks den Weg passierbar machen.
Dieser läuft unaufhaltsam von links nach rechts, dargestellt durch einen scrollenden Hintergrund. Leider ist das etwas ruckartig ausgefallen, wird aber durch die sehr gute und farbige Grafik wettgemacht.



BUILDERLAND – Bahnen Sie Ihrem Helden den Weg

Lemmings verkehrt

Sie reparieren deffekte Treppen, sprengen Durchgänge in Mauern, beseitigen Tretminen oder sorgen für die seltsamsten Fortbewegungsmittel. Ständig ist man auf der Suche nach Hilfsmitteln, damit der Held weiterkommt.
Hiermit ist aber das Spiel noch lange nicht ausgeschöpft. So hat unser Held natürlich auch Feinde, vor denen er geschützt werden muß. Ist dies nicht möglich, so kann er auch einmal dazu gebracht werden, Eigeninitiative zu entwickeln. Auf dem bisherigen Weg aufgesammelte Waffen und Gegenstände können ihm zur Verteidigung übergeben werden.
Auch können sich vermeintliche Feinde einmal als nützlich erweisen. So werden beispielsweise gefährliche Insekten zur Überquerung eines Hindernisses benutzt. Vorher müssen diese natürlich ausgetrickst werden.
Auf dem Weg trifft man auch häufig auf unvollständige Häuser. Baut man diese wieder auf, gelangt man in einen Bonuslevel. Hier verhält sich unser Held plötzlich ganz anders. Nun muß er mit dem Joystik durch ein einfaches Labyrinth bewegt werden. Ziel ist es, ein gefährlich aussehendes Monster zu erledigen. Hat man dies geschafft, erhält man für das weitere Spiel einen Zeitbonus.
Diese Bonuslevels sind sehr einfach zu bezwingen und stellen eher eine Erholung gegenüber dem Hauptspiel dar.
Eine Hintergrundmusik gibt es nicht. Dafür wurde bei dem Spielablauf nicht mit Toneffekten gespart. So hört man ständig die Schritte unseres Helden. Auch andere Gegenstände machen bei ihrer Bewegung entsprechende Geräusche, ohne aber mit der Zeit aufdringlich zu wirken.
Eine kurze Spielanleitung lag leider nur in Französisch und Englisch vor. Jedem, der gern Taktik und Geschicklichkeit in einem hat, kann man dieses Spiel empfehlen. Leider gibt es für dieses Spiel noch keinen Vertrieb in Deutschland. Aber LORICIEL versicherte uns, daß das Spiel auch bei ihnen direkt bestellt werden kann.

BUILDERLAND

Hersteller: LORICIEL
Steuerung: Joystick
Voraussetzung: für alle CPCs
Preis: etwa 60 DM
Bezugsquelle: LORICIEL

Bewertung:
Grafik: 1 — Sound: 2
Präsentation: 2 — Motivation: 1
Endnote: 2

Wir hoffen, daß wir Ihnen mit unserer Spiele-Review wieder einen Einblick in den derzeitigen Spielemarkt geben konnten. Vielleicht ist das eine oder andere Spielchen dabei, bei dem Sie sagen: "So etwas habe ich doch schon lange gesucht, und das gibt es also auch für den CPC!" Dann haben wir doch mal wieder für den CPC eine Bresche geschlagen und bewiesen, daß er "in" ist.
Jetzt noch etwas anderes: Da gibt es doch einige Freaks, die sich darüber beschweren, daß in unserer Review auch ältere Spiele vorgestellt werden. Sie sollten aber nicht vergessen, daß manche älteren Spiele besser gelungen sind als die sogenannten "brandneuen", und deshalb noch nichts an Reiz eingebüßt haben.
Einige neue Spiele stellen, betrachtet man die Sache genau, auch nur einen Abklatsch eines alten Spieles dar. Es mußte vielleicht gerade mal wieder ein neues auf den Markt, und dem Autor fiel nichts weiter ein, als irgendeinen Film zur Grundlage zu nehmen. Das Spiel bleibt immer das gleiche, nur die Figuren und Räume sehen etwas anders aus.
Für Neueinsteiger und "Nicht-Spiele-Freaks" soll diese Rubrik einen Überblick geben, was so auf dem Markt zu haben ist.
Viele Leser wünschen sich auch, daß in der Review gleich alle möglichen Tips und Tricks zu den Spielen gegeben werden. Wir meinen aber, daß man nicht gleich am Anfang eines Spieles alle Geheimnisse kennen sollte. Wo bliebe denn dann der Reiz des Spieles? Alle, die aber Schwierigkeiten haben, und irgendein Level eines Spieles ein unüberwindliches Hindernis darstellt, möchten wir an unsere Gamers Message verweisen, die jetzt wieder in jedem Heft zu finden ist.
Die in unserer Spiele-Review jeweils vorgestellten Spiele sind natürlich nicht nur bei den hier genannten, sondern auch bei anderen Händlern zu bekommen, nur irgendwoher müssen wir sie ja auch nehmen.
Nun genug der Worte und "Gut Spiel".

Andreas Lober/jg

Die Muster stellten uns zur Verfügung:

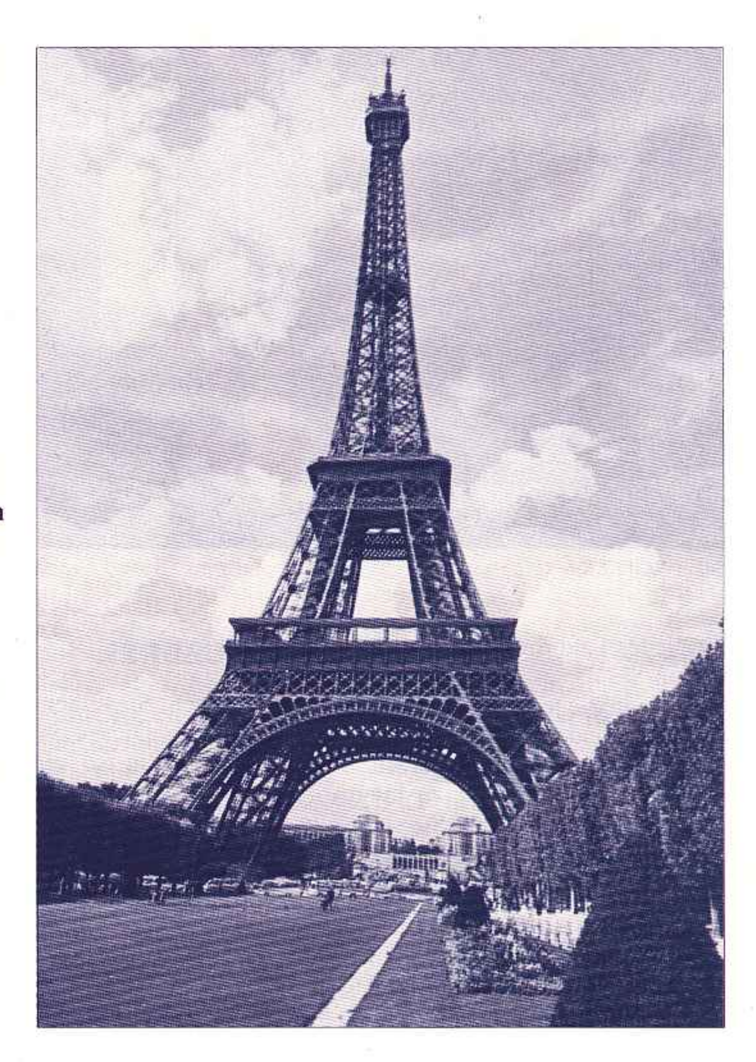
EDV-Obermeier
Bünder Straße 20
4972 Löhne
Tel.: 05732/6126

LORICIEL
81, rue de la Procession
92500 Rueil
Tel.: (0033) 1 47521183

Die angegebenen Preise beziehen sich, soweit nichts anderes angegeben ist, auf die Disketten-Version und die unter der Bezugsquelle angegebene Firma. Preise anderer Händler können davon abweichen.

Neues aus Frankreich

Vier Spitzenprogramme im Test



Schnell machte Jörg die Tür zu: "Ist etwa der Chefred in der Nähe?" Sein Blick löste sich von der Tür und schweifte langsam in die Richtung eines Paketes mit der Aufschrift "Loriciel". War dies etwa die Reaktion von Loriciel auf das kurz zuvor geführte Gespräch? Eigentlich mußte es so sein.
"Laß uns auspacken, Ralf, die versprochenen Spiele aus Frankreich sind da". Der Angesprochene hatte bereits den verträumten Blick aufgesetzt, flüsterte verzückt "Fronkraisch, Fronkraisch..." und schaltete schon mal den CPC an.

Tja, da lagen Sie vor uns auf dem Tisch: vier brandaktuelle Programme für die kleinen Amstrad-Rechner. Die Spiel-Orgie konnte also losgehen.

Super Skweek

Als erstes fiel unser Blick auf das Programm Super Skweek. Hierbei handelt es sich um ein Spiel, das nicht allein durch seine gute Grafik, sondern vielmehr durch das gut durchdachte Spielprinzip auffällt. Es geht darum, eine Spielfläche, die in mehrere Felder unterteilt ist, durch einmaliges Betreten selbiger einzufärben. Daß dies natürlich nicht gerade ohne Probleme vonstatten geht, versteht sich von selbst. Auf der Spielfläche laufen nämlich ganz nebenbei auch noch eine ganze Menge übler Burschen herum, wobei es nicht gerade einfach ist, diesen auszuweichen oder aber diese zu erschießen. Während das Programm in den ersten Leveln nur in einer Ebene spielt, kann man später mittels eines Lifts in andere Ebenen springen, in welchen man die gleiche Aufgabe hat. Dies kann sehr haarig werden, da hier nicht die ganzen Felder ausgefüllt sein müssen. So hat man teilweise nur schmale Laufstege, die am Ende wieder zu einem Lift führen. Kommt nun auch noch ein Gegner, der sich nicht abschießen läßt, heißt es, schnell zurückrennen und dem Gegner Platz machen.

Selbstverständlich kann man auch einen Laden besuchen, in dem man sich für das zwischendurch eingesammelte Geld neue Zusätze kaufen kann. Die Palette reicht von Munition, die in vier Richtungen losgeht, bis hin zu Bonusleben.

Auf jeden Fall ist dieses Spiel eine Betrachtung wert. Einziger Nachteil hierbei ist, daß leider nur eine französische Anleitung beiliegt. Nach einigen Testspielen wird man jedoch auch so herausgefunden haben, wie das Spiel funktioniert. Wer ein Actionstrategieballerspiel gesucht hat, wird hiermit sicherlich seine Freude haben.

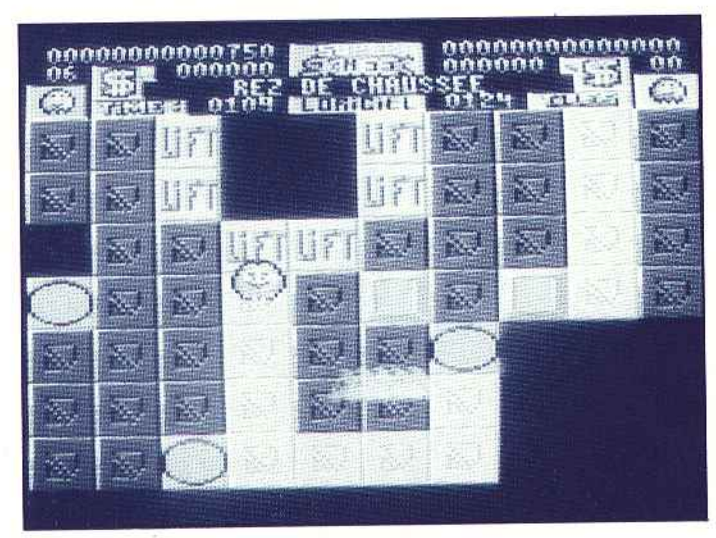
BOOLY

Welches sind Ihre Lieblingsspiele? Sie haben etwas für Taktikspiele, wo aber trotzdem etwas Geschicklichkeit gefragt ist, übrig? Dann ist *BOOLY* genau das richtige für Sie. Mehrere Personen können sich ebenfalls an diesem Spiel beteiligen. Die Aufgabe des Spieles besteht darin, einem Gitter aus Feldern ein einheitliches Aussehen zu geben. Vorgabe hierfür ist das Feld "GOOD" im Menü. Sie wählen das entsprechende Feld an, dieses dreht sich und wechselt damit die Farbe.

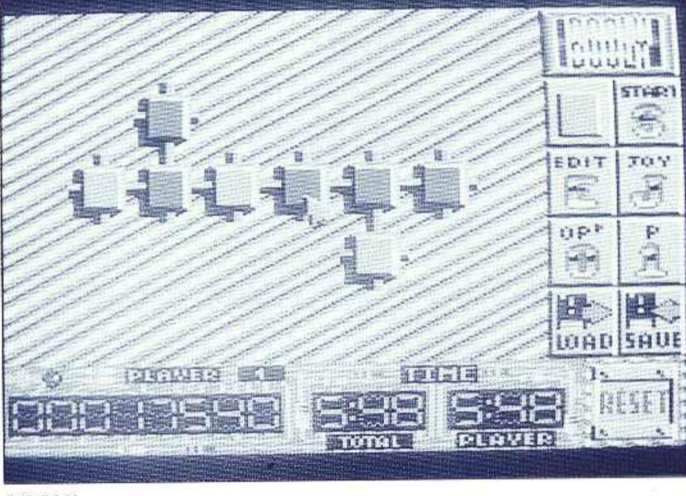
Damit aber nicht genug, denn einige Felder sind auch untereinander verbunden. Drehen Sie nun ein Feld, bewegen sich gleich einige umliegende mit. Dies macht schnell vorangegangene Bemühungen zunichte, was aber erst den Reiz des Spieles ausmacht. Zur Unterstützung können die Verbindungen der Felder auch sichtbar gemacht werden. Während dieser Darstellung läuft aber die Zeit um ein Vielfaches schneller ab. Mit dieser Option sollte man also sehr sparsam umgehen.

Haben Sie nun die Lösung für eine waagerechte, senkrechte oder diagonale Linie gefunden, so erhalten Sie einen Bonus.

Nach jeweils drei Bildern kommen Sie in die Spielstufe CHALLENGES. Hier liegt die Aufgabe ähnlich, doch kann



Super SKWEEK



BOOLY

hier jedes Feld drei Farben annehmen. Haben Sie dieses Bild geschafft, kommen Sie in die jeweils nächste Spielstufe. Als besonders nützlich ist der eingebaute Editor zu bezeichnen. Sind Ihnen die vorhandenen Bilder zu langweilig geworden, so "stricken" Sie sich selbst welche zusammen.

Mit dem Spiel erhalten Sie eine deutsche Anleitung, die aber leider nicht auf die Spielvariante für den CPC eingeht. So muß man doch ein wenig herumrätseln bis man herausgefunden hat, daß die Feuertaste durch Drücken aller vier Cursortasten erreicht wird, eigentlich eine sehr unübliche Tastenbelegung. Nun ist man in der Lage, die Bedienung des Spiels auf den Joystick umzuschalten. Für Nutzer eines Grün- oder Schwarzweiß-Monitors ist dieses Spiel weniger gut geeignet. Ebenfalls als etwas störend empfindet man die jeweils am Spielanfang stattfindende Handbuchabfrage.

Kennen Sie das Farbtheorem? Da wird behauptet, daß man jede beliebig unterteilte Fläche mit vier Farben füllen kann. Das klingt ja relativ einfach. Aber es sollen niemals zwei Farben nebeneinander liegen! Sie glauben, schnell das Gegenteil beweisen zu können?



JE DECOUVRE LES FORMES

QUADREL

Starten Sie einen Versuch mit *QUADREL*. Dieses Spiel macht es Ihnen aber nicht so einfach wie eben beschrieben. Hier haben Sie nur eine begrenzte Anzahl jeder Farbe zur Verfügung. Es gilt, eine Strategie zu entwickeln, um alle Flächen mit einer Farbe zu füllen. Aber aufgepaßt, denn der Computer gestattet Ihnen erst gar nicht, zwei gleiche Farben nebeneinander zu plazieren.
Das Programm steht in vier Varianten zur Verfügung. Erstens, Sie spielen frei und allein. Zweitens Sie spielen allein, aber der Computer schreibt Ihnen vor, welches Feld Sie als nächstes mit einer Farbe füllen sollen. Die Farbe können Sie, soweit vorhanden, frei wählen. Drittens, Sie spielen gegen den Computer und viertens gegen einen menschlichen Gegenspieler. Eine Zeitbegrenzung, an der das Spiel abgebrochen wird, gibt es nicht. Dagegen spielt aber die Zeit bei der Punkteberechnung eine Rolle. Also wenn Sie einen Gegenspieler haben, sollten Sie sich schon etwas beeilen. Da es bei dem Spiel um Farben geht, ist natürlich ein Color-Monitor Voraussetzung.

Für die Kleinen

Wenn man Kinder hat, kennt man das leidige Problem: Auch die allerkleinsten wollen schon mal mit dem Computer spielen. Meist hat man aber kein passendes Spiel, welches schon unsere Jüngsten bewältigen könnten. In diese Lücke drängt sich nun LORICIEL mit ihrem Spiel *JE DECOUVRE LES FORMES*.
Das klingt völlig französisch, aber keine Angst: Das Spiel ist für Kinder von drei bis acht Jahren geeignet und kommt völlig ohne Schrift und Sprache aus. Wie auch ein des Französischen Unkundiger dem Namen entnehmen kann, beschäftigt sich dieses Spiel mit Formen.
Kurz gesagt ist es ein Puzzle. Man kann zwischen insgesamt 24 Bildern aus drei Kategorien wählen.
Hat man ein Bild gewählt, so hat man jetzt noch die Möglichkeit festzulegen, in wieviel Teile das Bild zerlegt werden soll. Für die unerfahrensten Spieler gibt es die Möglichkeit, mit fünf Teilen zu beginnen. Alte Hasen können sich dagegen an 80 Teilen messen. Letztere Aufgabe ist schon reichlich kompliziert, so daß diese Variante schon für Puzzlefreunde unter den Erwachsenen interessant wird. Mit einer weiteren Option wird bestimmt, ob die Teile vermischt werden sollen oder in ihrer richtigen Reihenfolge zur Bildzusammensetzung zur Verfügung stehen.
Dieses Spiel wird nur mit einer französischen Kurzanleitung geliefert. Aber das ist kein eigentliches Manko, da es sehr leicht zu bedienen ist.
Die vorgestellten Spiele sind alle von der Firma LORICIEL aus Frankreich. In Deutschland hat sich bis heute leider noch kein Vertreiber für diese Spiele gefunden. Über aktuelle Bezugsmöglichkeiten sollten Sie sich direkt bei der Firma informieren. Die Adresse haben wir in einer Box abgedruckt. Jedes Spiel kostet 199 FF, das sind etwa sechzig DM.

rs/jg

LORICIEL S.A.
81, rue de la Procession
92500 Rueil-Malmaison
Tel.: (1) 47 52 11 33

"Englisches Wetter" für die CPCs



Ein Blick auf die "Business Computing'91"

Was hat denn ein Bericht über eine Computermesse in England in unserem Magazin zu suchen? Viele werden sich diese Frage wohl stellen. Aber besonders für die PCW-Anwender haben wir da eine Überraschung: Dort in London wurde der neue PCW 9512 plus vorgestellt.

Vom 17. bis 20. September fand in London eine Computermesse, die "Business Computing '91", statt. Alle renommierten Hard- und Softwarehäuser gaben sich hier ein Stelldichein. Als britischer Hersteller mit Gespür für den Inlandsmarkt war selbstverständlich auch Amstrad vertreten. Die Computer-Abteilung des Elektronik-Konzerns präsentierte dem interessierten Publikum hier die System-Palette für die kommenden Monate. Wir nutzten die Gelegenheit, um den Amstrad-Vertretern am Stand ein paar Fragen zu stellen und den einen oder anderen Blick auf die gezeigten Neuheiten zu erhaschen.

Bekannte Technik, neu gemischt

Wer den Stand aufmerksam betrachtete, dem fiel ein Computer ins Auge, der im Design etwas von den anderen abwich. Monitor und Computer bilden bei diesem Gerät eine Einheit – es ist auch deutlich kleiner als die üblichen PCs. Der untere Teil ist jedoch etwas abgesetzt, so daß man von weitem den Eindruck hat, der Monitor würde auf dem Computer stehen. Sieht man jedoch genauer hin, so erkennt man ein eingebautes 3,5 Zoll-Laufwerk. Ja, aber da hatten wir ihn doch, den PCW 9512 plus. Waren unsere Informationen also doch richtig – der Acht-Bit-Rechner ist noch nicht out. Die Technik des "Neuen" ist ganz so, wie wir es bereits in unserem Vorab-Bericht in Heft 10/11'91 (Seite 117) beschrieben haben. Also: in puncto Prozessor, Bildschirm, Tastatur und Arbeitsprinzip nichts Neues. Aber: ein neuer Tintenstrahldrucker, eine veränderte Gehäusearchitektur und natürlich das "endlich kompatible" Diskettenformat mit 720 kByte auf 3,5 Zoll.

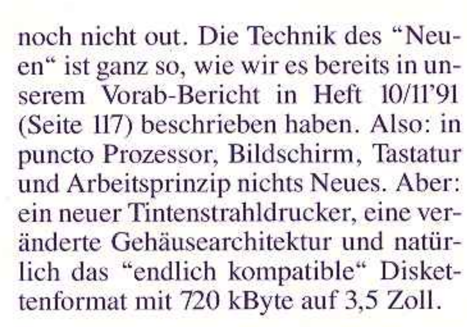
Bei Amstrad wies man besonders auf das Preis-Leistungsverhältnis der neuen PCWs hin – aha, die Zielgruppe dürfte also bei den wenig finanzkräftigen Schülern, Studenten und privaten Haushalten zu suchen sein.

Immerhin liegt mit den PCWs ein wirklich ausgereiftes Computersystem vor. Jahrelange Arbeit von Soft- und Hardware-Entwicklern hat sich bezahlt gemacht. Die spezielle Locomotive-Software wird ständig weiterentwickelt. Eine Festplatte läßt sich nachrüsten, in den Plus sogar intern einbauen. Auch an Hardwarezusätzen für anspruchsvolle Anwendungen fehlt es nicht. So ist es heute möglich, mit einem PCW – egal welcher Baureihe – Texte und Grafiken mit einem Komfort zu verarbeiten, der auf einem 8-Bit-System seinesgleichen sucht. Bilder lassen sich einscannen, Grafiken in Texte einbinden und die fertigen Seiten direkt vom Computer aus als Fax versenden. Was hat ein um ein Vielfaches teurerer High-Tech-PC in dieser Beziehung mehr zu bieten?

Und der CPC?

Nun zu einer etwas traurigeren Angelegenheit. Leider äußerte Amstrad auf der Messe auf unser Anfragen hin, daß sie keine Zukunft mehr für die CPC-Rechner sehen. Leider betrachtet man dort den CPC als reinen Spielcomputer. Daß dieses System auch noch zu mehr fähig ist, scheint Amstrad völlig entgangen zu sein.
So wurde als Nachfolger der CPC-plus-Serie ein IBM-kompatibler "Spiel-PC" vorgestellt. Dieser hat den Aufbau eines 80286-Rechners mit einem Megabyte RAM und verfügt über VGA-Farbgrafik. Eine serienmäßig eingebaute "Ad Lib"-kompatible Soundkarte sowie zwei optisch äußerst unpassende Mini-Boxen im Eier-Look sorgen für musikalische Begleitung. Außerdem gibt es einen Joystick-Anschluß samt zugehörigem Joystick. Eine Festplatte ist eingebaut. Auf dieser werden mit dem Kauf des Geräts gleich einige Spiele mitgeliefert, die sich dann nach dem Start sogleich über ein Menü auswählen lassen.
Irgendwie kommt man dabei doch auf die Idee, daß Amstrad mit diesem Computer dem Amiga Konkurrenz machen will. Nur: ob das mit dieser Ausstattung und dem Preis von 800 Pfund (etwa 2400 DM) gelingt, scheint zumindest fraglich. Ein CPC erzeugt für wesentlich weniger Geld auch keine schlechteren Geräusche als die "Ad Lib", und ob man zum Spielen wirklich einen IBM-kompatiblen Rechner mit all seinen teuren Kompromissen und System-Hintertürchen braucht, darf getrost bezweifelt werden.
Wir bleiben trotz "Schlechtwetterfront" dem CPC und dem PCW treu. Sie doch hoffentlich auch?

jg



Links der PCW 9512+ mit Tintenstrahldrucker, rechts der Spiel-PC Amstrad PC5286

RSX-Professional: eine gigantische BASIC-Erweiterung mit Wumm

Der Hammer

Schaut man sich die für PC-Computer geschriebenen BASIC-Compiler an, muß man doch dem Locomotive-BASIC auf dem CPC so manche Mängel anhaften. Daß sich diese jedoch in Form neuer RSX-Routinen leicht beheben lassen, zeigt diese Bibliothek mit 35 neuen Befehlen. Sie hilft dem BASIC-Interpreter des CPC in allen Lebenslagen trickreich auf die Sprünge.

Neben Befehlen zur Pull-Down-Menü-Programmierung enthält RSX-Professional mehrere Befehle zur Verwaltung der zweiten 64 kByte des CPC 6128. Aber auch das Arbeiten mit Speicherblöcken sowie Bildschirmausschnitten wird wesentlich vereinfacht. Alle Befehle, die auf den Bildschirm zugreifen unterstützen zudem die drei Modi des CPC. Aufgrund der vielen Speicherfunktionen läuft das Programm nur mit mindestens 128 kByte Speicher. Die selbst erstellten Programme sollten übrigens maximal sieben kByte groß sein, da es sonst zu Überschneidungen kommen kann. Als kleine Demonstration, daß auch mit nur sieben kByte wirklich gute Programme entwickelt werden können, sollten Sie sich die drei Demonstrationsprogramme anschauen, welche die Leistung von RSX-Professional gut aufzeigen.

Pull-Down-Funktionen

Kommen wir zuerst zu den Befehlen, die sich mit der Pull-Down-Menü-Programmierung beschäftigen.

OPENWINDOW,X,Y,Breite,Höhe,"Dachzeile","Fußzeile"
Diese Routine öffnet an der Position X, Y ein Fenster und speichert den überschriebenen Bildschirminhalt an einer anderen Stelle im Speicher ab. Somit gehen beim Schließen des Fensters keine wichtigen Bildschirminformationen verloren. Die hier zu übergebenden Parameter müssen anhand des jeweiligen Textmodus errechnet werden. Wenn ein Fehler auftritt, sollte der Computer dies anzeigen.

CLOSEWINDOW
Schließt das zuletzt geöffnete Fenster oder Menü und holt den gesicherten Hintergrund wieder aus dem Speicher zurück.

CLEARWINDOWS
Beläßt alle geöffneten Fenster und Menüs auf dem Bildschirm, löscht jedoch alle gesicherten Hintergründe.

GETMEM, Adresse
Übergibt die Größe des noch freien Speicherplatzes, der für den Fensterhintergrund reserviert wurde an die angegebene Adresse.
Um die noch freie Speicherkapazität für den Hintergrund zu ermitteln, sollten Sie folgende Formel benutzen:

```
X=PEEK(Adresse+1) 256+PEEK( Adresse)
```

Hierbei muß *Adresse* jedoch keine Variable sein, sondern kann von Anfang an fest eingestellt werden.

TEXT,X,Y,Text Stift, Text Hintergrund, "Text"
Dieser Befehl ähnelt der PRINT-Anweisung von BASIC. Er wird jedoch schneller abgearbeitet und bietet die Möglichkeit, gleich die Koordinaten sowie die Stift- und Papier-Farben festzulegen.

SETMODE,Modus
Setzt den Grafikmodus fest, in dem alle Kommandos abgearbeitet werden sollen. Wird als Parameter eine Drei übergeben, so wird automatisch der jeweils aktuelle Modus übernommen.

SETBORDER, Stift, Hintergrund, Randart
Wählt die Umrandungsart sowie die Farbe für den Stift (Pen) und den Hintergrund (Paper) aus. Sollte keine Umrandung erwünscht sein, geben Sie bei Randart einfach eine Null ein. Bei Eingabe von Eins wird die Umrandung durch eine Linie dargestellt und bei Eingabe von Drei erscheint ein Schachbrettmuster als Umrandung.

NAME, Stift für oberen Titel, Untergrund für oberen Titel

SUBNAME, Stift, Untergrund
Gleiche Funktion wie *NAME* , lediglich für den unteren Fenstertitel.

INSIDE, Hintergrundfarbe für Fenster

DEFAULT
Alle bisher geänderten Parameter werden wieder zurückgesetzt.

MENU,"1. Option", "2. Option", ..., "letzte Option", X, Y, "Überschrift", "Untertitel"
Öffnet ein Auswahlfenster mit den angegebenen Optionen. Die Anzahl der Menüpunkte ist von drei bis 23 frei einstellbar. Auch hier wird der Hintergrund gesichert.

MENUONLY <gleiche Parameter wie bei der Funktion zuvor>
Erfüllt die gleiche Funktion wie *MENU*, der Hintergrund wird jedoch nicht gesichert.

LINEMENU,"1. Option","2. Option",...,"letzte Option",Y
Öffnet in der angegebenen Zeile ein Zeilenmenü. Der Hintergrund wird hierbei nicht gesichert.

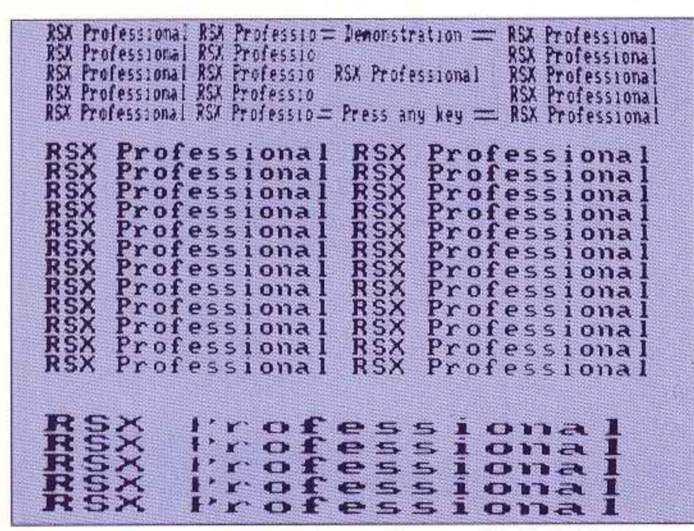
DELOPTION, Option
Gibt an, welcher Menüpunkt gesperrt werden soll.

PUTOPTION, Option
Ein zuvor gesperrter Menüpunkt wird jetzt wieder freigegeben.

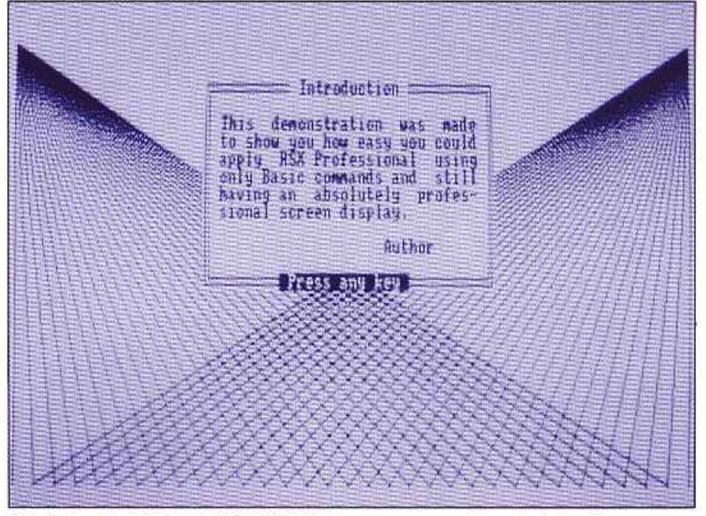
RESETOPTIONS
Alle Menüpunkte werden wieder zur Auswahl freigegeben.

TEXTCOLOR, Menüpunktstift, -hintergrund, Stift für Cursorbalken
Setzt die Farben für ein Auswahlfenster.

MENUCOMMANDS, Links, Rechts, Hoch, Runtern, Auswahl, Menü schließen
Gibt an, mit welchen Tasten die genannten Funktionen ausgeführt werden. Hier müssen Sie die ASCII-Codes eingeben. Als Defaultwerte wurden die



Die Befehlserweiterung RSX-Professional kann in allen drei Bildschirmmodi gleichzeitig arbeiten.



Wird der Befehl CLOSEWINDOW nun angewandt, wird das Fenster geschlossen und der ursprüngliche Hintergrund erscheint wieder.

Cursortasten, sowie ENTER und ESC voreingestellt.

Die Interruptverwaltung

SETINTERRUPT, erste Adresse, zweite Adresse, ...
Legt eine beliebige Anzahl von Maschinencode-Programmen fest, die in der angegebenen Reihenfolge aufgerufen werden sollen. Diese Programme sollten an einer beliebigen Adresse stehen und werden nacheinander abgearbeitet.
Sie müssen die Register nicht selbständig mittels *PUSH* und *POP* sichern. Lediglich sollten Sie beachten, daß die Programme nie länger als 1/300 Sekunde dauern und auf jeden Fall mit einem *RET* abgeschlossen sind.
FASTINTERRUPT
Selektiert den Interrupt, der 300 mal innerhalb einer Sekunde aufgerufen werden soll.
SLOWINTERRUPT
Selektiert einen Interrupt, der fünfzig mal pro Sekunde ausgeführt werden soll.
INTERRUPTOFF
Schaltet den gewählten Interrupt aus.

Eine RAM-Disk muß her

Als kleinen Bonus für all die, die die zweiten 64 kByte anderweitig verwenden möchten, bietet RSX-Professionell noch einige Befehle zur Verwaltung einer RAM-Disk mit beliebiger Größe.
RAMDISK, Größe
Erstellt eine RAM-Disk mit einer in kBytes anzugebenden Größe. Maximal sollten Sie sich jedoch auf 62 kBytes beschränken.
RAMDISKOFF
Schaltet die RAM-Disk ab. Alle Befehle um die RAM-Disk werden jetzt nicht mehr ausgeführt und brechen mit einer Fehlermeldung ab.
SAVE, "name", Startadresse, Länge
Sichert einen bestimmten Speicherbereich auf der RAM-Disk. Der Name der Datei kann maximal zehn Buchstaben lang sein. Ebenfalls wird eine Unterscheidung zwischen Klein- und Großschreibung vorgenommen.
LOAD, "Name", Startadresse
Lädt eine zuvor auf der RAM-Disk gesicherte Datei an die angegebene Startadresse im Speicher. Wird keine Startadresse an gegeben, wird die beim Abspeichern angegebene benutzt.
DELETE, "Name"
Löscht eine Datei von der RAM-Disk. Wenn ein x angegeben wurde, werden alle Dateien gelöscht.
RENAME, "Neu", "Alt"
Benennt eine Datei um.
CAT
Zeigt das Inhaltsverzeichnis der RAM-Disk auf dem Bildschirm an.

Speichermanipulationen

LDIR, Originaladresse, Neue Adresse, Länge
Kopiert einen Speicherbereich mit beliebiger Länge an eine neue Adresse.
LDDR
Dieser Befehl wird mit den gleichen Parametern wie zuvor aufgerufen. Weitere Informationen hierzu entnehmen Sie bitte eventuell vorhandener Literatur zum Thema Maschinensprache.

FILL, Füllbyte, Anfangsadresse, Blocklänge
Füllt einen Speicherblock mit dem vorzugebendem Füllbyte.

GETIMAGE, X, Y, Breite, Höhe, Adresse
Kopiert einen Teil des Bildschirms (Sprite) an eine bestimmte Adresse im Speicher. X und Y sind hierbei Byte-Werte (60 200). Die Gesamtlänge des Sprites im Speicher entspricht XxY+2.

PUTIMAGE, Modus, X, Y, Adresse
Nimmt ein Sprite aus dem Speicher und bringt es an der angegebenen Position auf den Bildschirm. Bei Modus können Sie einen Wert von Null bis Drei eintragen.

Null benutzt keinerlei logischen Funktionen und bringt das Sprite wie im Original auf den Bildschirm.

Eins benutzt die OR-Anweisung. *Wenn schon ein Punkt gesetzt ist, oder wenn ein Punkt gesetzt werden soll, bleibt der Punkt bestehen oder wird gesetzt.*

Zwei benutzt die AND-Anweisung. *Wenn an der entsprechenden Position schon ein Punkt gesetzt ist und ein Punkt gesetzt werden soll, nur dann bleibt der Punkt bestehen.*

Drei benutzt die XOR-Anweisung. *Es darf nur eine Bedingung wahr sein. Ist ein Punkt gesetzt* **oder** *soll ein Punkt gesetzt werden, so wird er auch gesetzt.*

Damir Petkovic/rs

```
10 'RSXPRO.BAS                                [703]
20 'Installiert die Basicerweiterung          [1143]
30 '(c) 1991 Damir Petkovic & CPC Internat [2092]
ional
40 INK 2,15:INK 0,26:INK 1,0:BORDER 26:MOD [2181]
E 2
50 MEMORY &87FF                               [738]
60 LOAD"!rsxpro.bin",&8800                    [1561]
70 CALL &8800                                 [485]
80 |OPENWINDOW,23,7,33,11,""," Press any k [2613]
ey "
90 |TEXT,31,9,1,0,"RSX Professional"          [3328]
100 |TEXT,26,11,1,0,"This program was orig [4187]
inally"
110 |TEXT,27,12,1,0,"(c) 1991 Damir Petkov [1857]
ic"
120 |TEXT,26,15,1,0,"  & CPC International [2487]
"
130 CALL &BB06                                [393]
140 |CLOSEWINDOW:NEW                          [1190]
```

```
1000 'RSXPRO.LAD                              [686]
1001 'Generiert RSXPRO.BIN                    [2259]
1002 '(c) 1991 Damir Petkovic & CPC Intern [2092]
ational
```

```
1003 MEMORY &87FF                             [738]
1004 FOR adr=&8800 TO &A01D STEP 8            [1881]
1005   FOR i=adr TO adr+ 7                    [1011]
1006     READ b$                              [315]
1007     byte=VAL("&"+b$)                     [465]
1008     POKE i,byte                          [294]
1009   NEXT i                                 [375]
1010 NEXT adr                                 [547]
1011 SAVE"rsxpro.bin",b,&8800,&181D           [1350]
1012 END                                      [110]
1013 DATA AF,CD,1E,9C,AF,CD,36,8D             [1245]
1014 DATA 21,A2,89,01,11,88,C3,D1             [1445]
1015 DATA BC,82,88,C3,6F,91,C3,6A             [908]
1016 DATA 93,C3,F4,93,C3,FB,93,C3             [1238]
1017 DATA 1A,94,C3,36,8D,C3,8A,8D             [1055]
1018 DATA C3,98,8D,C3,B6,8D,C3,CC             [1504]
1019 DATA 8D,C3,E2,8D,C3,68,95,C3             [1321]
1020 DATA 62,95,C3,8D,94,C3,20,98             [874]
1021 DATA C3,B9,94,C3,B3,94,C3,7C             [1418]
1022 DATA 8D,C3,E2,94,C3,13,95,C3             [1079]
1023 DATA 50,97,C3,AE,97,C3,B9,97             [972]
1024 DATA C3,07,98,C3,12,9A,C3,1D             [984]
1025 DATA 9A,C3,28,9A,C3,6A,9A,C3             [994]
1026 DATA CB,9A,C3,40,9C,C3,3B,9D             [1459]
1027 DATA C3,26,9E,C3,CD,9E,C3,66             [1501]
1028 DATA 9F,C3,03,A0,C3,BB,9B,C3             [604]
1029 DATA 1E,9C,4F,50,45,4E,57,49             [1398]
1030 DATA 4E,44,4F,D7,43,4C,4F,53             [865]
1031 DATA 45,57,49,4E,44,4F,D7,43             [1221]
1032 DATA 4C,45,41,52,57,49,4E,44             [1499]
1033 DATA 4F,57,D3,47,45,54,4D,45             [1932]
1034 DATA CD,54,45,58,D4,44,45,46             [875]
1035 DATA 41,55,4C,D4,53,45,54,4D             [885]
1036 DATA 4F,44,C5,53,45,54,42,4F             [584]
```

```
1037 DATA 52,44,45,D2,4E,41,4D,C5          [1167]
1038 DATA 53,55,42,4E,41,4D,C5,49          [1327]
1039 DATA 4E,53,49,44,C5,4D,45,4E          [1392]
1040 DATA D5,4D,45,4E,55,4F,4E,4C          [1516]
1041 DATA D9,4D,45,4E,55,49,4E,50          [1264]
1042 DATA 55,D4,4C,49,4E,45,4D,45          [1266]
1043 DATA 4E,D5,44,45,4C,4F,50,54          [893]
1044 DATA 49,4F,CE,50,55,54,4F,50          [980]
1045 DATA 54,49,4F,CE,52,45,53,45          [1262]
1046 DATA 54,4F,50,54,49,4F,4E,D3          [1207]
1047 DATA 54,45,58,54,43,4F,4C,4F          [848]
1048 DATA D2,4D,45,4E,55,43,4F,4D          [1085]
1049 DATA 4D,41,4E,44,D3,53,45,54          [1598]
1050 DATA 49,4E,54,45,52,52,55,50          [1411]
1051 DATA D4,46,41,53,54,49,4E,54          [1208]
1052 DATA 45,52,52,55,50,D4,53,4C          [571]
1053 DATA 4F,57,49,4E,54,45,52,52          [1560]
1054 DATA 55,50,D4,49,4E,54,45,52          [1166]
1055 DATA 52,55,50,54,4F,46,C6,4C          [651]
1056 DATA 44,49,D2,4C,44,44,D2,46          [1260]
1057 DATA 49,4C,CC,47,45,54,49,4D          [1145]
1058 DATA 41,47,C5,50,55,54,49,4D          [1391]
1059 DATA 41,47,C5,53,41,56,C5,4C          [1068]
1060 DATA 4F,41,C4,44,45,4C,45,54          [1564]
1061 DATA C5,52,45,4E,41,4D,C5,43          [945]
1062 DATA 41,D4,43,41,54,4D,45,CD          [1802]
1063 DATA 52,41,4D,44,49,53,CB,52          [912]
1064 DATA 41,4D,44,49,53,4B,4F,46          [1298]
1065 DATA C6,00,00,00,00,00,00,00          [1156]
1066 DATA 00,00,00,00,00,00,00,00          [1005]
1067 DATA 00,00,00,00,00,00,00,00          [1005]
1068 DATA 00,00,00,00,00,00,00,00          [1005]
1069 DATA 00,00,00,00,00,00,00,00          [1005]
1070 DATA 00,00,00,00,00,00,00,00          [1005]
1071 DATA 00,00,00,00,00,00,00,00          [1005]
1072 DATA 03,00,01,02,00,01,00,01          [1021]
1073 DATA 00,00,00,00,00,00,00,00          [1005]
1074 DATA 00,00,00,00,00,00,00,00          [1005]
1075 DATA 00,00,00,00,00,00,00,00          [1005]
1076 DATA 00,00,00,DA,8E,1D,8F,87          [1343]
1077 DATA 8F,00,00,00,00,00,00,00          [177]
1078 DATA 00,00,00,00,00,00,00,00          [1005]
1079 DATA 00,00,00,00,00,00,00,00          [1005]
1080 DATA 00,00,00,4E,6F,20,6E,61          [1260]
1081 DATA 6D,65,0D,0A,52,41,4D,44          [846]
1082 DATA 49,53,4B,20,20,63,61,74          [1647]
1083 DATA 61,6C,6F,67,75,65,0D,0A          [1262]
1084 DATA 0A,20,20,4B,20,66,72,65          [1505]
1085 DATA 65,0D,0A,0A,00,00,00,00          [499]
1086 DATA 00,00,00,00,00,00,00,00          [1005]
1087 DATA 00,00,00,00,00,00,00,00          [1005]
1088 DATA 00,00,00,00,00,00,00,00          [1005]
1089 DATA 00,00,00,00,00,00,00,00          [1005]
1090 DATA 00,00,00,00,00,00,00,00          [1005]
1091 DATA 00,00,00,00,00,00,00,00          [1005]
1092 DATA 00,00,00,00,00,00,00,00          [1005]
1093 DATA 50,72,6F,67,72,61,6D,6D          [1141]
1094 DATA 65,64,20,62,79,20,44,61          [1652]
1095 DATA 6D,69,72,20,50,65,74,6B          [1405]
1096 DATA 6F,76,69,63,20,6F,6E,20          [1095]
1097 DATA 32,38,2E,30,38,2E,31,39          [1271]
1098 DATA 39,30,2E,00,00,00,00,00          [1245]
1099 DATA 00,00,00,00,00,00,00,00          [1005]
1100 DATA 00,00,00,00,00,00,00,00          [1005]
1101 DATA 00,00,00,00,00,00,00,00          [1005]
1102 DATA 00,00,00,00,00,00,00,00          [1005]
1103 DATA 00,00,00,00,00,00,00,00          [1005]
1104 DATA 00,00,00,00,00,00,00,00          [1005]
1105 DATA 00,00,00,00,00,00,00,00          [1005]
1106 DATA 00,00,00,00,00,00,00,00          [1005]
1107 DATA 00,00,00,00,00,00,00,00          [1005]
1108 DATA 00,00,00,00,00,00,00,00          [1005]
1109 DATA 00,00,00,1F,1F,18,18,18          [953]
1110 DATA 00,00,00,F8,F8,18,18,18          [809]
1111 DATA 18,18,18,1F,1F,00,00,00          [1007]
1112 DATA 18,18,18,F8,F8,00,00,00          [839]
1113 DATA 18,18,18,18,18,18,18,18          [1528]
1114 DATA 00,00,00,FF,FF,00,00,00          [1113]
1115 DATA 00,00,3F,20,20,27,24,24          [1053]
1116 DATA 00,00,FC,04,04,E4,24,24          [1254]
1117 DATA 24,24,27,20,20,3F,00,00          [1213]
1118 DATA 24,24,E4,04,04,FC,00,00          [1499]
1119 DATA 24,24,24,24,24,24,24,24          [1316]
1120 DATA 00,00,FF,00,00,FF,00,00          [550]
1121 DATA AA,55,AA,55,AA,55,AA,55          [868]
1122 DATA 00,00,00,00,00,00,00,00          [1005]
1123 DATA 2D,2D,2D,2D,2D,2D,20,21          [1116]
1124 DATA 22,23,24,25,26,27,28,29          [1535]
1125 DATA 2A,2B,2C,2C,2C,2C,2C,2C          [1141]
1126 DATA 00,00,00,00,00,00,00,00          [1005]
1127 DATA 00,C0,0C,CC,30,F0,3C,FC          [1814]
1128 DATA 03,C3,0F,CF,33,F3,3F,FF          [903]
1129 DATA 00,F0,0F,FF,00,F0,0F,FF          [958]
1130 DATA 00,F0,0F,FF,00,F0,0F,FF          [958]
1131 DATA 00,FF,00,FF,00,FF,00,FF          [1120]
1132 DATA 00,FF,00,FF,00,FF,00,FF          [1120]
1133 DATA 07,45,72,72,6F,72,20,69          [1438]
```

```
1134 DATA 6E,20,70,61,72,61,6D,65          [1743]
1135 DATA 74,65,72,73,20,21,00,07          [1196]
1136 DATA 54,6F,6F,20,6D,61,6E,79          [1562]
1137 DATA 20,77,69,6E,64,6F,77,73          [1280]
1138 DATA 20,21,00,07,4E,6F,20,77          [1284]
1139 DATA 69,6E,64,6F,77,73,20,6F          [1315]
1140 DATA 70,65,6E,20,21,00,07,57          [1403]
1141 DATA 69,6E,64,6F,77,20,74,69          [859]
1142 DATA 74,6C,65,20,74,6F,6F,20          [1527]
1143 DATA 6C,6F,6E,67,20,21,00,07          [1951]
1144 DATA 57,69,6E,64,6F,77,20,73          [998]
1145 DATA 75,62,2D,74,69,74,6C,65          [973]
1146 DATA 20,74,6F,6F,20,6C,6F,6E          [1319]
1147 DATA 67,20,21,00,07,57,69,6E          [592]
1148 DATA 64,6F,77,20,6F,75,74,20          [767]
1149 DATA 6F,66,20,73,63,72,65,65          [1739]
1150 DATA 6E,20,21,00,07,4F,6E,65          [1448]
1151 DATA 20,69,6E,74,65,72,72,75          [1581]
1152 DATA 70,74,20,69,73,20,61,6C          [1271]
1153 DATA 72,65,61,64,79,20,4F,4E          [850]
1154 DATA 20,21,00,07,49,6E,74,65          [1553]
1155 DATA 72,72,75,70,74,20,72,6F          [665]
1156 DATA 75,74,69,6E,65,20,4D,55          [1231]
1157 DATA 53,54,20,4E,4F,54,20,62          [1293]
1158 DATA 65,20,69,6E,20,72,61,6E          [1540]
1159 DATA 67,65,20,62,65,74,77,65          [1086]
1160 DATA 65,6E,20,26,34,30,30,30          [1648]
1161 DATA 20,61,6E,64,20,26,38,30          [1092]
1162 DATA 30,30,20,21,00,07,46,69          [1522]
1163 DATA 6C,65,6E,61,6D,65,20,74          [1294]
1164 DATA 6F,6F,20,6C,6F,6E,67,20          [1657]
1165 DATA 21,00,07,46,69,6C,65,20          [1090]
1166 DATA 61,6C,72,65,61,64,79,20          [1366]
1167 DATA 65,78,69,73,74,73,20,21          [1543]
1168 DATA 00,07,52,41,4D,44,49,53          [1209]
1169 DATA 4B,20,66,75,6C,6C,20,21          [1807]
1170 DATA 00,07,46,69,6C,65,20,6E          [1199]
1171 DATA 6F,74,20,66,6F,75,6E,64          [1738]
1172 DATA 20,21,00,07,52,41,4D,44          [579]
1173 DATA 49,53,4B,20,6E,6F,74,20          [1147]
1174 DATA 72,65,61,64,79,20,21,00          [1180]
1175 DATA 07,4C,49,4E,45,20,4D,45          [1439]
1176 DATA 4E,55,20,74,6F,6F,20,6C          [1506]
1177 DATA 6F,6E,67,20,21,00,21,FF          [1450]
1178 DATA FF,22,A9,89,21,00,00,22          [582]
1179 DATA A7,89,22,B2,89,C9,B7,C2          [1452]
1180 DATA B8,8E,21,D8,89,11,E1,89          [1720]
1181 DATA 01,09,00,ED,B0,AF,CD,07          [1097]
1182 DATA 98,AF,32,BD,89,3C,32,BC          [1240]
1183 DATA 89,32,B8,89,C6,02,32,BE          [1306]
1184 DATA 89,21,00,BF,22,B9,89,36          [835]
1185 DATA 00,3E,FC,32,63,96,3E,0D          [1473]
1186 DATA 32,72,96,3E,F0,32,77,96          [874]
1187 DATA 3C,32,7C,96,3C,32,68,96          [1342]
1188 DATA 3C,32,6D,96,21,E8,8A,11          [1940]
1189 DATA E9,8A,01,16,00,36,00,ED          [1049]
1190 DATA B0,C9,FE,01,C2,B8,8E,DD          [1590]
1191 DATA 7E,00,E6,03,32,E1,89,C9          [1845]
1192 DATA FE,03,C2,B8,8E,DD,7E,00          [1786]
1193 DATA E6,03,32,E4,89,DD,7E,02          [1836]
1194 DATA E6,0F,32,E3,89,DD,7E,04          [1483]
1195 DATA E6,0F,32,E2,89,C9,FE,02          [1093]
1196 DATA C2,B8,8E,DD,7E,00,E6,0F          [601]
1197 DATA 32,E6,89,DD,7E,02,E6,0F          [1096]
1198 DATA 32,E5,89,C9,FE,02,C2,B8          [1828]
1199 DATA 8E,DD,7E,00,E6,0F,32,E8          [1253]
1200 DATA 89,DD,7E,02,E6,0F,32,E7          [467]
1201 DATA 89,C9,FE,01,C2,B8,8E,DD          [1558]
1202 DATA 7E,00,E6,0F,32,E9,89,C9          [1219]
1203 DATA CD,06,B9,E1,11,88,8B,01          [1211]
1204 DATA 08,00,ED,B0,CD,09,B9,00          [1688]
1205 DATA 00,00,E1,11,88,8B,01,08          [1175]
1206 DATA 00,ED,B0,00,00,00,3A,A6          [903]
1207 DATA 89,47,B7,3E,14,C8,87,10          [994]
1208 DATA FD,C9,3A,A6,89,B7,47,3E          [1034]
1209 DATA 04,C8,CB,3F,10,FC,C9,CD          [1146]
1210 DATA 1A,8E,47,AF,86,10,FD,C9          [1302]
1211 DATA 21,AE,89,CD,27,8E,6F,26          [1138]
1212 DATA 00,E5,3A,AF,89,47,21,00          [2089]
1213 DATA 00,11,50,00,B7,28,03,19          [1482]
1214 DATA 10,FD,D1,19,11,00,C0,19          [1778]
1215 DATA C9,7A,E6,C0,07,07,F6,C4          [1494]
1216 DATA 06,7F,4F,ED,49,CB,BA,CB          [775]
1217 DATA F2,C9,2A,A7,89,06,06,2B          [1248]
1218 DATA 10,FD,22,A7,89,21,D7,8B          [612]
1219 DATA 18,49,21,EB,8B,18,44,21          [829]
1220 DATA FE,8B,18,3F,21,17,8C,18          [1318]
1221 DATA 3A,21,34,8C,18,35,21,4C          [1095]
1222 DATA 8C,18,30,21,6B,8C,18,2B          [1806]
1223 DATA 21,AD,8C,18,26,01,C0,7F          [1499]
1224 DATA ED,49,21,C2,8C,18,1C,21          [1222]
1225 DATA D9,8C,18,17,01,C0,7F,ED          [1618]
1226 DATA 49,21,E9,8C,18,0D,21,FB          [819]
1227 DATA 8C,18,08,21,10,8D,18,03          [1503]
1228 DATA 21,C0,8B,7E,B7,C8,CD,5A          [1952]
1229 DATA BB,23,18,F7,F5,3A,A6,89          [1432]
1230 DATA 87,87,87,87,6F,26,00,11          [1313]
```

```
1231 DATA 90,8B,19,F1,5F,16,00,19          [1153]
1232 DATA 7E,C9,3E,08,F5,E5,3E,7F          [1505]
1233 DATA 32,ED,8E,3E,77,32,F7,8E          [852]
1234 DATA 3E,04,F5,1A,CB,7F,79,20          [1376]
1235 DATA 01,78,E6,AA,77,1A,CB,77          [1080]
1236 DATA 79,20,01,78,E6,55,B6,77          [1160]
1237 DATA 3A,ED,8E,D6,10,32,ED,8E          [1759]
1238 DATA D6,08,32,F7,8E,23,F1,3D          [1293]
1239 DATA 20,D8,13,E1,7C,C6,08,67          [1307]
1240 DATA F1,3D,20,C0,C9,3E,08,F5          [1404]
1241 DATA E5,3E,7F,32,3A,8F,3E,77          [1370]
1242 DATA 32,44,8F,3E,6F,32,4F,8F          [1536]
1243 DATA 3E,67,32,5A,8F,3E,02,F5          [1368]
1244 DATA 1A,CB,7F,79,20,01,78,E6          [2005]
1245 DATA 88,77,1A,CB,77,79,20,01          [1495]
1246 DATA 78,E6,44,B6,77,1A,CB,6F          [469]
1247 DATA 79,20,01,78,E6,22,B6,77          [1311]
1248 DATA 1A,CB,67,79,20,01,78,E6          [1483]
1249 DATA 11,B6,77,3E,5F,32,3A,8F          [1280]
1250 DATA 3E,57,32,44,8F,3E,4F,32          [766]
1251 DATA 4F,8F,3E,47,32,5A,8F,23          [1517]
1252 DATA F1,3D,20,BB,13,E1,7C,C6          [1687]
1253 DATA 08,67,F1,3D,20,99,C9,79          [1418]
1254 DATA EE,FF,32,91,8F,06,08,1A          [844]
1255 DATA EE,00,77,3E,08,84,67,13          [749]
1256 DATA 10,F5,C9,E5,3A,A6,89,87          [1723]
1257 DATA 5F,16,00,21,FB,89,19,5E          [1537]
1258 DATA 23,56,ED,53,F0,8F,2A,AE          [1242]
1259 DATA 89,E5,ED,43,AE,89,CD,30          [2064]
1260 DATA 8E,D1,ED,53,AE,89,D1,EB          [1596]
1261 DATA 7E,B7,C8,E5,D6,20,6F,26          [1621]
1262 DATA 00,29,29,29,ED,4B,F1,89          [887]
1263 DATA 09,D5,E5,CD,06,B9,E1,11          [1339]
1264 DATA 88,8B,01,08,00,ED,B0,CD          [2018]
1265 DATA 09,B9,3A,B4,89,47,3A,B5          [989]
1266 DATA 89,4F,E1,11,88,8B,E5,CD          [1019]
1267 DATA DA,8E,CD,1A,8E,D1,6F,26          [1147]
1268 DATA 00,19,EB,E1,23,18,C1,21          [1478]
1269 DATA F5,89,22,28,91,22,08,91          [1338]
1270 DATA 21,F7,89,22,05,91,21,EA          [1476]
1271 DATA 89,22,19,91,21,EB,89,22          [615]
1272 DATA 1F,91,AF,32,D1,90,3C,32          [742]
1273 DATA E3,90,CD,AF,90,2A,F1,89          [654]
1274 DATA 22,F3,89,21,00,8B,22,F1          [897]
1275 DATA 89,DD,7E,04,21,97,8A,77          [1540]
1276 DATA 23,36,00,2B,3A,B1,89,D6          [1630]
1277 DATA 02,ED,4B,AE,89,04,E5,C5          [2074]
1278 DATA F5,CD,9B,8F,F1,C1,E1,04          [1876]
1279 DATA 3D,20,F3,3A,B0,89,3D,ED          [1931]
1280 DATA 4B,AE,89,81,4F,04,3A,B1          [2308]
1281 DATA 89,D6,02,E5,C5,F5,CD,9B          [1017]
1282 DATA 8F,F1,C1,E1,04,3D,20,F3          [1239]
1283 DATA 2A,F3,89,22,F1,89,2A,AE          [1385]
1284 DATA 89,E5,3A,B1,89,3D,84,67          [1415]
1285 DATA 22,AE,89,21,F6,89,22,28          [1560]
1286 DATA 91,22,08,91,21,F9,89,22          [1939]
1287 DATA 05,91,21,EC,89,22,19,91          [1272]
1288 DATA 21,ED,89,22,1F,91,3E,02          [1527]
1289 DATA 32,D1,90,3C,32,E3,90,CD          [1326]
1290 DATA AF,90,E1,22,AE,89,C9,3A          [1401]
1291 DATA EE,89,32,B5,89,3A,EF,89          [1147]
1292 DATA 32,B4,89,3A,F0,89,47,87          [1364]
1293 DATA 80,87,6F,26,00,11,70,8B          [669]
1294 DATA 19,E5,DD,E1,21,97,8A,DD          [1528]
1295 DATA 7E,00,77,23,3A,B0,89,D6          [1270]
1296 DATA 02,47,DD,7E,05,77,23,10          [1307]
1297 DATA FC,DD,7E,01,77,23,36,00          [1261]
1298 DATA 2A,F1,89,22,F3,89,21,00          [1429]
1299 DATA 8B,22,F1,89,ED,4B,AE,89          [1073]
1300 DATA 21,97,8A,CD,9B,8F,2A,F3          [963]
1301 DATA 89,22,F1,89,2A,F7,89,3A          [682]
1302 DATA F5,89,B7,28,2C,4F,06,00          [1102]
1303 DATA 11,97,8A,ED,B0,EB,36,00          [1487]
1304 DATA 3A,EA,89,32,B5,89,3A,EB          [1402]
1305 DATA 89,32,B4,89,3A,B0,89,21          [852]
1306 DATA F5,89,96,CB,3F,ED,4B,AE          [2073]
1307 DATA 89,81,4F,21,97,8A,CD,9B          [847]
1308 DATA 8F,3A,EE,89,32,B5,89,3A          [1213]
1309 DATA EF,89,32,B4,89,C9,FE,03          [1113]
1310 DATA D4,11,BC,32,A6,89,3E,20          [1275]
1311 DATA CD,A5,BB,E5,21,F0,8D,30          [1294]
1312 DATA 03,21,FF,8D,11,D3,8F,01          [1290]
1313 DATA 0F,00,ED,B0,E1,22,F1,89          [1361]
1314 DATA CD,0E,8E,32,AD,89,C9,FE          [1493]
1315 DATA 06,C2,B8,8E,3A,E1,89,CD          [1323]
1316 DATA 46,91,3E,03,DD,CB,04,7E          [601]
1317 DATA C2,B8,8E,DD,BE,04,D2,B8          [558]
1318 DATA 8E,DD,CB,06,7E,C2,B8,8E          [1038]
1319 DATA DD,BE,06,D2,B8,8E,AF,DD          [2030]
1320 DATA BE,05,C2,B8,8E,DD,BE,07          [1717]
1321 DATA C2,B8,8E,DD,BE,09,C2,81          [635]
1322 DATA 8E,DD,BE,0B,C2,81,8E,DD          [670]
1323 DATA 6E,02,DD,66,03,DD,7E,06          [951]
1324 DATA BE,DA,77,8E,7E,32,F5,89          [1360]
1325 DATA 23,5E,23,56,ED,53,F7,89          [1457]
1326 DATA DD,6E,00,DD,66,01,DD,7E          [1477]
1327 DATA 06,BE,DA,7C,8E,7E,32,F6          [1741]
```

```
1328 DATA 89,23,5E,23,56,ED,53,F9          [1585]
1329 DATA 89,21,AD,89,DD,7E,0A,BE          [687]
1330 DATA D2,81,8E,DD,7E,06,3D,BE          [1445]
1331 DATA D2,81,8E,DD,86,0A,BE,D2          [2100]
1332 DATA 81,8E,DD,7E,08,FE,19,D2          [1793]
1333 DATA 81,8E,DD,7E,04,3D,FE,19          [2040]
1334 DATA D2,81,8E,DD,86,08,FE,19          [1490]
1335 DATA D2,81,8E,DD,7E,04,32,B1          [1800]
1336 DATA 89,DD,7E,06,32,B0,89,DD          [1212]
1337 DATA 7E,08,32,AF,89,DD,7E,0A          [664]
1338 DATA 32,AE,89,21,B0,89,CD,27          [1093]
1339 DATA 8E,32,AC,89,ED,5B,A7,89          [1741]
1340 DATA 01,C4,7F,ED,49,CB,F2,D5          [2060]
1341 DATA CD,30,8E,EB,E1,73,23,72          [1576]
1342 DATA 23,3A,AC,89,77,23,3A,B1          [1489]
1343 DATA 89,87,87,87,77,23,ED,5B          [1254]
1344 DATA A9,89,73,23,72,23,CB,B4          [1241]
1345 DATA 22,A7,89,01,C0,7F,ED,49          [1066]
1346 DATA 5F,16,00,6A,62,3A,AC,89          [981]
1347 DATA 47,19,10,FD,EB,2A,A9,89          [747]
1348 DATA A7,ED,52,DA,62,8E,3A,A8          [1296]
1349 DATA 89,4F,7C,B9,DA,62,8E,20          [632]
1350 DATA 07,3A,A7,89,BD,D2,62,8E          [1621]
1351 DATA 22,A9,89,23,E5,CD,30,8E          [937]
1352 DATA D1,CD,51,8E,32,AB,89,3A          [1759]
1353 DATA B1,89,87,87,87,47,C5,E5          [1015]
1354 DATA 3A,AC,89,ED,A0,CB,7A,20          [1765]
1355 DATA 18,3D,20,F7,E1,01,00,08          [1629]
1356 DATA 09,30,04,01,50,C0,09,C1          [1308]
1357 DATA 10,E4,01,C0,7F,ED,49,18          [2204]
1358 DATA 14,F5,3A,AB,89,3C,32,AB          [1371]
1359 DATA 89,4F,06,7F,ED,49,CB,BA          [649]
1360 DATA CB,F2,F1,18,D4,3A,E7,89          [1632]
1361 DATA CD,C4,8E,32,EC,89,3A,E8          [1230]
1362 DATA 89,CD,C4,8E,32,ED,89,3A          [1475]
1363 DATA E5,89,CD,C4,8E,32,EA,89          [1135]
1364 DATA 3A,E6,89,CD,C4,8E,32,EB          [2064]
1365 DATA 89,3A,E4,89,32,F0,89,3A          [815]
1366 DATA E9,89,CD,C4,8E,32,4C,93          [942]
1367 DATA 3A,E2,89,CD,C4,8E,32,EE          [1141]
1368 DATA 89,3A,E3,89,CD,C4,8E,32          [555]
1369 DATA EF,89,CD,FF,8F,21,AE,89          [1978]
1370 DATA 34,23,34,CD,30,8E,E5,21          [1441]
1371 DATA B0,89,35,35,CD,27,8E,32          [1139]
1372 DATA 48,93,3A,B1,89,D6,02,87          [1336]
1373 DATA 87,87,E1,E5,5D,54,13,01          [1032]
1374 DATA 00,00,0B,36,00,ED,B0,E1          [1455]
1375 DATA 01,00,08,09,30,04,01,50          [1060]
1376 DATA C0,09,3D,20,E6,2A,B2,89          [1961]
1377 DATA 23,22,B2,89,3E,FF,32,BF          [2034]
1378 DATA 89,C9,B7,C2,B8,8E,2A,B2          [1511]
1379 DATA 89,7D,B4,CA,72,8E,2B,22          [1783]
1380 DATA B2,89,01,C4,7F,ED,49,2A          [2313]
1381 DATA A7,89,CB,F4,2B,56,2B,5E          [700]
1382 DATA 2B,ED,53,A9,89,13,7E,32          [1575]
1383 DATA BD,93,47,2B,7E,32,C1,93          [848]
1384 DATA 2B,E5,D5,5F,16,00,21,00          [1826]
1385 DATA 00,19,10,FD,EB,E1,A7,ED          [1280]
1386 DATA 52,D1,E5,EB,56,2B,5E,CB          [1126]
1387 DATA B4,22,A7,89,EB,D1,CD,51          [1100]
1388 DATA 8E,32,AB,89,06,00,C5,E5          [1509]
1389 DATA 06,00,1A,77,23,13,CB,7A          [1281]
1390 DATA 20,16,10,F6,E1,01,00,08          [767]
1391 DATA 09,30,04,01,50,C0,09,C1          [1308]
1392 DATA 10,E4,01,C0,7F,ED,49,C9          [1964]
1393 DATA C5,3A,AB,89,3C,32,AB,89          [1632]
1394 DATA 4F,06,7F,ED,49,CB,BA,CB          [1208]
1395 DATA F2,C1,18,D6,B7,C2,B8,8E          [1108]
1396 DATA C3,26,8D,3D,C2,B8,8E,2A          [1743]
1397 DATA A9,89,23,ED,5B,A7,89,06          [2078]
1398 DATA 06,13,10,FD,A7,ED,52,EB          [1428]
1399 DATA DD,6E,00,DD,66,01,73,23          [1061]
1400 DATA 72,C9,FE,05,C2,B8,8E,3A          [1785]
1401 DATA E1,89,CD,46,91,21,AD,89          [1320]
1402 DATA DD,7E,08,BE,D2,B8,8E,DD          [555]
1403 DATA 7E,06,FE,19,D2,B8,8E,AF          [1817]
1404 DATA DD,BE,09,C2,B8,8E,DD,BE          [2300]
1405 DATA 07,C2,B8,8E,DD,6E,00,DD          [1611]
1406 DATA 66,01,7E,DD,86,08,21,AD          [1560]
1407 DATA 89,BE,D2,B8,8E,DD,6E,00          [813]
1408 DATA DD,66,01,4E,23,5E,23,56          [1245]
1409 DATA EB,06,00,11,97,8A,ED,B0          [984]
1410 DATA EB,36,00,DD,7E,02,E6,0F          [1569]
1411 DATA CD,C4,8E,32,B4,89,DD,7E          [1306]
1412 DATA 04,E6,0F,CD,C4,8E,32,B5          [1511]
1413 DATA 89,DD,46,06,DD,4E,08,21          [1101]
1414 DATA 97,8A,C3,9B,8F,FE,02,C2          [1070]
1415 DATA B8,8E,DD,7E,01,B7,C2,B8          [1559]
1416 DATA 8E,DD,7E,00,FE,01,DA,B8          [990]
1417 DATA 8E,FE,18,D2,B8,8E,32,B8          [891]
1418 DATA 89,DD,6E,02,DD,66,03,22          [1110]
1419 DATA B9,89,C9,F5,AF,32,DB,94          [1603]
1420 DATA F1,FE,01,C2,B8,8E,DD,7E          [2341]
1421 DATA 01,B7,C2,B8,8E,DD,7E,00          [1893]
1422 DATA FE,18,D2,B8,8E,FE,01,DA          [1152]
1423 DATA B8,8E,5F,16,00,1B,21,E8          [1129]
1424 DATA 8A,19,36,FF,3E,FF,32,DB          [1804]
```

```
1425 DATA 94,C9,FE,03,C2,B8,8E,AF     [1435]
1426 DATA DD,BE,01,C2,B8,8E,DD,BE     [1619]
1427 DATA 03,C2,B8,8E,DD,BE,05,C2     [2063]
1428 DATA B8,8E,DD,7E,00,E6,0F,32     [1380]
1429 DATA BE,89,DD,7E,02,E6,0F,32     [1830]
1430 DATA BD,89,DD,7E,04,E6,0F,32     [1576]
1431 DATA BC,89,C9,FE,06,C2,B8,8E     [763]
1432 DATA AF,DD,BE,01,C2,B8,8E,DD     [1195]
1433 DATA BE,03,C2,B8,8E,DD,BE,05     [894]
1434 DATA C2,B8,8E,DD,BE,07,C2,B8     [351]
1435 DATA 8E,DD,BE,09,C2,B8,8E,DD     [887]
1436 DATA BE,0B,C2,B8,8E,DD,7E,00     [1260]
1437 DATA 32,63,96,DD,7E,02,32,72     [1043]
1438 DATA 96,DD,7E,04,32,7C,96,DD     [686]
1439 DATA 7E,06,32,77,96,DD,7E,08     [1156]
1440 DATA 32,6D,96,DD,7E,0A,32,68     [1675]
1441 DATA 96,C9,21,64,93,22,CF,95     [1436]
1442 DATA FE,1C,D2,B8,8E,FE,05,DA     [1399]
1443 DATA B8,8E,F5,F5,DD,E5,E1,11     [1052]
1444 DATA C0,89,01,04,00,ED,B0,11     [1141]
1445 DATA C8,89,01,04,00,ED,B0,D6     [1989]
1446 DATA 02,32,C4,89,AF,32,C5,89     [1540]
1447 DATA 32,C7,89,F1,D6,04,47,FD     [1721]
1448 DATA 21,A5,95,3E,08,FD,77,02     [2075]
1449 DATA 3C,FD,77,05,AF,DD,6E,08     [828]
1450 DATA DD,66,09,BE,30,01,7E,FD     [836]
1451 DATA 34,02,FD,34,02,FD,34,05     [1401]
1452 DATA FD,34,05,10,E8,C6,04,32     [1507]
1453 DATA C6,89,AF,32,BF,89,DD,E5     [1192]
1454 DATA DD,21,C0,89,3E,06,CD,6F     [866]
1455 DATA 91,21,6F,91,22,CF,95,DD     [1558]
1456 DATA E1,F1,21,BF,89,CB,7E,C8     [1324]
1457 DATA F5,3A,BC,89,CD,C4,8E,32     [1550]
1458 DATA B5,89,3A,BD,89,CD,C4,8E     [1701]
1459 DATA 32,B4,89,3A,BE,89,CD,C4     [2156]
1460 DATA 8E,32,3B,97,3A,C4,89,21     [1083]
1461 DATA C8,89,86,D6,02,32,B7,89     [1458]
1462 DATA 3A,CA,89,C6,02,32,B6,89     [902]
1463 DATA F1,D6,04,32,BB,89,47,FD     [1045]
1464 DATA 21,24,96,3E,08,FD,77,02     [1078]
1465 DATA 3C,FD,77,05,DD,6E,08,DD     [890]
1466 DATA 66,09,7E,B7,28,1C,C5,4F     [1929]
1467 DATA 06,00,23,5E,23,56,EB,11     [1208]
1468 DATA 97,8A,ED,B0,EB,36,00,ED     [1256]
1469 DATA 4B,B6,89,21,97,8A,CD,9B     [937]
1470 DATA 8F,C1,21,B7,89,35,FD,34     [1116]
1471 DATA 02,FD,34,02,FD,34,05,FD     [958]
1472 DATA 34,05,10,C8,CD,E3,96,CD     [1562]
1473 DATA 06,BB,FE,FC,CC,85,96,FE     [1613]
1474 DATA F2,CC,85,96,FE,F3,CC,85     [718]
1475 DATA 96,FE,0D,CC,8E,96,FE,F0     [1410]
1476 DATA CC,9A,96,FE,F1,CC,CF,96     [2385]
1477 DATA CD,03,BB,18,DA,2A,B9,89     [1493]
1478 DATA 77,CD,0E,97,E1,C9,CD,0E     [1738]
1479 DATA 97,2A,B9,89,3A,B8,89,77     [992]
1480 DATA E1,C9,CD,0E,97,3A,B8,89     [1675]
1481 DATA 3D,20,03,3A,BB,89,32,B8     [1012]
1482 DATA 89,3A,BB,89,47,3A,B8,89     [995]
1483 DATA 5F,16,00,1B,21,E8,8A,19     [1308]
1484 DATA 7E,B7,28,52,3A,B8,89,3D     [1746]
1485 DATA 20,03,3A,BB,89,32,B8,89     [1537]
1486 DATA 10,E3,CD,06,BB,E1,C9,CD     [1570]
1487 DATA 0E,97,3A,B8,89,3C,21,BB     [1663]
1488 DATA 89,BE,38,04,28,02,3E,01     [1188]
1489 DATA 32,B8,89,3A,BB,89,47,3A     [1375]
1490 DATA B8,89,5F,16,00,1B,21,E8     [1471]
1491 DATA 8A,19,7E,B7,28,18,3A,B8     [1380]
1492 DATA 89,3C,21,BB,89,BE,38,04     [1012]
1493 DATA 28,02,3E,01,32,B8,89,10     [1187]
1494 DATA DE,CD,06,BB,E1,C9,3A,B8     [1637]
1495 DATA 89,21,C8,89,86,32,AF,89     [804]
1496 DATA 3A,CA,89,C6,02,32,AE,89     [1184]
1497 DATA CD,30,8E,E5,3A,C6,89,D6     [944]
1498 DATA 04,21,97,8A,77,CD,27,8E     [990]
1499 DATA 32,39,97,E1,06,08,C5,E5     [868]
1500 DATA 06,00,3E,00,AE,77,23,10     [1670]
1501 DATA F9,E1,01,00,08,09,30,04     [901]
1502 DATA 01,50,C0,09,C1,10,E7,C9     [1377]
1503 DATA FE,29,D2,B8,8E,FE,01,DA     [1076]
1504 DATA B8,8E,4F,3A,01,8A,B7,C2     [1530]
1505 DATA 86,8E,79,21,45,8A,22,02     [916]
1506 DATA 8A,11,46,8A,01,51,00,36     [905]
1507 DATA 00,ED,B0,47,3D,87,5F,16     [1209]
1508 DATA 00,21,45,8A,19,FD,21,89     [862]
1509 DATA 97,AF,FD,77,02,3C,FD,77     [1089]
1510 DATA 05,DD,5E,00,DD,56,01,73     [1426]
1511 DATA 23,72,7A,E6,C0,FE,40,CA     [1438]
1512 DATA 8B,8E,2B,2B,2B,FD,34,02     [1774]
1513 DATA FD,34,02,FD,34,05,FD,34     [1288]
1514 DATA 05,10,DE,C3,C4,97,B7,C2     [1163]
1515 DATA B8,8E,21,C4,97,22,AC,97     [1505]
1516 DATA C9,B7,C2,B8,8E,21,D9,97     [1037]
1517 DATA 22,AC,97,C9,CD,19,BD,21     [1280]
1518 DATA CC,89,11,EA,97,01,00,80     [843]
1519 DATA CD,E0,BC,3E,02,32,01,8A     [2089]
1520 DATA C9,21,CC,89,11,EA,97,06     [1734]
1521 DATA 80,CD,D7,BC,3E,01,32,01     [1500]
```

```
1522 DATA 8A,C9,2A,02,8A,5E,23,56     [438]
1523 DATA EB,CD,06,98,2A,02,8A,23     [1134]
1524 DATA 23,23,7E,2B,B6,20,03,21     [1399]
1525 DATA 45,8A,22,02,8A,C9,E9,B7     [1140]
1526 DATA C2,B8,8E,3A,01,8A,B7,C8     [745]
1527 DATA 21,01,8A,36,00,21,CC,89     [992]
1528 DATA FE,01,CA,DD,BC,C3,E6,BC     [884]
1529 DATA FE,16,D2,B8,8E,FE,02,DA     [814]
1530 DATA B8,8E,F5,3A,E1,89,CD,46     [393]
1531 DATA 91,F1,21,97,8A,77,AF,DD     [1661]
1532 DATA BE,01,C2,B8,8E,DD,7E,00     [1523]
1533 DATA FE,19,D2,B8,8E,46,77,05     [1001]
1534 DATA FD,21,7B,98,78,32,C9,99     [1625]
1535 DATA 32,EF,99,87,FD,77,02,3C     [1491]
1536 DATA FD,77,05,C5,CD,0E,8E,C1     [414]
1537 DATA 32,DC,98,11,1C,A0,21,98     [1474]
1538 DATA 8A,36,20,23,36,20,23,36     [1296]
1539 DATA 20,21,10,A2,22,0E,A2,78     [1165]
1540 DATA 32,04,A4,DD,6E,02,DD,66     [1472]
1541 DATA 03,7E,FE,15,D2,B8,8E,C5     [1323]
1542 DATA D5,23,5E,23,56,EB,B7,20     [1805]
1543 DATA 05,21,99,8A,3E,01,D1,E5     [1339]
1544 DATA F5,6B,62,01,1C,A0,A7,ED     [1422]
1545 DATA 42,7D,2A,0E,A2,77,23,F1     [570]
1546 DATA 77,23,22,0E,A2,E1,4F,06     [1224]
1547 DATA 00,ED,B0,C1,C5,78,FE,01     [1322]
1548 DATA 28,14,21,98,8A,01,03,00     [847]
1549 DATA ED,B0,FD,35,02,FD,35,02     [1460]
1550 DATA FD,35,05,FD,35,05,C1,10     [1133]
1551 DATA AA,EB,36,00,11,1C,A0,A7     [1194]
1552 DATA ED,52,7D,FE,50,D2,B3,8E     [1532]
1553 DATA 3A,BC,89,CD,C4,8E,32,B5     [757]
1554 DATA 89,3A,BD,89,CD,C4,8E,32     [976]
1555 DATA B4,89,3A,BE,89,CD,C4,8E     [1002]
1556 DATA 32,BA,99,3A,97,8A,0E,00     [1920]
1557 DATA 47,21,1C,A0,CD,9B,8F,3A     [1087]
1558 DATA B8,89,3D,21,04,A4,BE,38     [1118]
1559 DATA 01,AF,3C,32,B8,89,CD,C8     [1080]
1560 DATA 99,CD,06,BB,21,68,96,BE     [1279]
1561 DATA 28,23,21,6D,96,BE,28,31     [1582]
1562 DATA 21,72,96,BE,28,41,21,63     [788]
1563 DATA 96,BE,28,46,21,77,96,BE     [928]
1564 DATA 28,40,21,7C,96,BE,28,3A     [1223]
1565 DATA CD,03,BB,18,D4,CD,81,99     [1596]
1566 DATA 3A,B8,89,3D,20,03,3A,04     [1782]
1567 DATA A4,32,B8,89,CD,EE,99,18     [759]
1568 DATA C0,CD,81,99,3A,B8,89,21     [1006]
1569 DATA 04,A4,BE,20,01,AF,3C,32     [1381]
1570 DATA B8,89,CD,C8,99,18,AA,CD     [1371]
1571 DATA 81,99,3A,B8,89,2A,B9,89     [1393]
1572 DATA 77,C9,2A,B9,89,77,CD,27     [1497]
1573 DATA 8E,3A,97,8A,87,87,87,5F     [1647]
1574 DATA 16,00,6F,62,29,29,19,29     [1085]
1575 DATA E5,3A,B8,89,3D,87,5F,16     [1132]
1576 DATA 00,21,10,A2,19,CD,27,8E     [1768]
1577 DATA F5,23,CD,27,8E,32,B7,99     [1307]
1578 DATA F1,D1,6F,26,00,19,11,00     [1343]
1579 DATA C0,19,06,08,C5,E5,06,00     [1364]
1580 DATA 7E,EE,00,77,23,10,F9,E1     [1266]
1581 DATA 3E,08,84,67,C1,10,ED,C9     [2137]
1582 DATA 06,00,3A,B8,89,3D,5F,16     [599]
1583 DATA 00,21,E8,8A,19,7E,B7,28     [1380]
1584 DATA A8,3A,B8,89,21,04,A4,BE     [871]
1585 DATA 20,01,AF,3C,32,B8,89,10     [998]
1586 DATA E1,CD,06,BB,E1,C9,06,00     [1301]
1587 DATA 3A,B8,89,3D,5F,16,00,21     [1082]
1588 DATA E8,8A,19,7E,B7,28,82,3A     [1730]
1589 DATA B8,89,3D,20,03,3A,04,A4     [1157]
1590 DATA 32,B8,89,10,E3,CD,06,BB     [1029]
1591 DATA E1,C9,FE,03,C2,B8,8E,CD     [1673]
1592 DATA 3E,9A,ED,B0,C9,FE,03,C2     [973]
1593 DATA B8,8E,CD,3E,9A,ED,B8,C9     [398]
1594 DATA FE,03,C2,B8,8E,CD,3E,9A     [1320]
1595 DATA EB,7A,B7,C2,B8,8E,73,5D     [1510]
1596 DATA 54,13,0B,ED,B0,C9,DD,4E     [1468]
1597 DATA 00,DD,46,01,DD,5E,02,DD     [1580]
1598 DATA 56,03,DD,6E,04,DD,66,05     [1032]
1599 DATA C9,4F,E6,F8,26,13,6F,B1     [1431]
1600 DATA AD,29,54,5D,14,14,8F,84     [949]
1601 DATA 67,79,29,29,19,48,06,00     [945]
1602 DATA 09,C9,FE,05,C2,B8,8E,AF     [426]
1603 DATA DD,BE,03,C2,B8,8E,DD,BE     [2049]
1604 DATA 05,C2,B8,8E,DD,BE,07,C2     [1142]
1605 DATA B8,8E,DD,BE,09,C2,B8,8E     [1016]
1606 DATA DD,7E,08,FE,50,D2,B8,8E     [1288]
1607 DATA 47,DD,7E,06,FE,C8,D2,B8     [1197]
1608 DATA 8E,CD,51,9A,DD,7E,04,32     [1255]
1609 DATA B8,9A,DD,46,02,DD,5E,00     [733]
1610 DATA DD,56,01,DD,7E,04,12,13     [953]
1611 DATA DD,7E,02,12,13,C5,E5,01     [581]
1612 DATA 00,00,ED,B0,E1,01,00,08     [1313]
1613 DATA 09,30,04,01,50,C0,09,C1     [1308]
1614 DATA 10,EB,C9,FE,04,C2,B8,8E     [1246]
1615 DATA AF,DD,BE,03,C2,B8,8E,DD     [1110]
1616 DATA BE,05,C2,B8,8E,DD,BE,07     [1717]
1617 DATA C2,B8,8E,DD,7E,06,E6,03     [939]
1618 DATA 5F,16,00,21,2F,9B,19,7E     [834]
```

```
1619 DATA 32,1A,9B,DD,7E,04,FE,50          [1007]
1620 DATA D2,B8,8E,47,DD,7E,02,FE          [1568]
1621 DATA C8,D2,B8,8E,CD,51,9A,DD          [2574]
1622 DATA 5E,00,DD,56,01,1A,32,18          [1686]
1623 DATA 9B,13,1A,13,47,C5,E5,06          [1304]
1624 DATA 00,1A,B6,77,13,23,10,F9          [1277]
1625 DATA E1,01,00,08,09,30,04,01          [1085]
1626 DATA 50,C0,09,C1,10,E7,C9,00          [1757]
1627 DATA B6,A6,AE,21,10,8A,11,11          [963]
1628 DATA 8A,01,09,00,36,20,ED,B0          [953]
1629 DATA C9,EB,11,10,8A,4F,06,00          [975]
1630 DATA ED,B0,C9,2A,06,8A,11,40          [1702]
1631 DATA 00,19,EB,CD,51,8E,EB,FD          [1456]
1632 DATA 21,10,8A,E5,EB,2A,04,8A          [597]
1633 DATA CB,BC,CB,F4,2B,C1,A7,ED          [1247]
1634 DATA 42,D8,EB,4B,42,FD,7E,00          [557]
1635 DATA ED,B1,F5,2B,F1,28,02,37          [1565]
1636 DATA C9,FD,23,FD,7E,00,B7,20          [1143]
1637 DATA 08,11,09,00,A7,ED,52,A7          [1395]
1638 DATA C9,23,FD,7E,00,BE,28,E9          [1311]
1639 DATA 18,C5,ED,5B,06,8A,CD,51          [1461]
1640 DATA 8E,EB,01,00,40,79,BE,28          [1202]
1641 DATA 05,0C,23,10,F9,3D,51,1E          [1196]
1642 DATA 00,CB,22,CB,22,C9,6F,26          [1743]
1643 DATA 00,5F,54,29,29,29,29,A7          [1348]
1644 DATA ED,52,C9,FE,01,C2,B8,8E          [1388]
1645 DATA DD,7E,01,B7,C2,B8,8E,DD          [1027]
1646 DATA 7E,00,FE,3F,D2,B8,8E,32          [1004]
1647 DATA 08,8A,3C,4F,CD,A6,9B,21          [1350]
1648 DATA 00,00,A7,ED,52,22,06,8A          [1205]
1649 DATA 2B,22,27,8D,11,41,00,19          [1850]
1650 DATA 22,04,8A,ED,5B,06,8A,CD          [1451]
1651 DATA 51,8E,6B,62,13,01,3F,00          [925]
1652 DATA 36,FF,ED,B0,EB,3A,08,8A          [1791]
1653 DATA B7,28,13,16,00,5F,A7,ED          [1726]
1654 DATA 52,36,00,3D,28,08,5D,54          [1065]
1655 DATA 13,4F,06,00,ED,B0,01,C0          [1097]
1656 DATA 7F,ED,49,C3,26,8D,B7,C2          [1338]
1657 DATA B8,8E,32,08,8A,21,FF,FF          [1581]
1658 DATA 22,27,8D,22,06,8A,22,04          [1525]
1659 DATA 8A,C3,26,8D,ED,5B,04,8A          [603]
1660 DATA 3E,FF,BA,C0,F1,C3,AE,8E          [1380]
1661 DATA FE,03,C2,B8,8E,CD,33,9B          [1589]
1662 DATA DD,6E,04,DD,66,05,7E,FE          [990]
1663 DATA 0B,D2,90,8E,23,5E,23,56          [572]
1664 DATA FE,01,20,08,F5,1A,FE,2A          [722]
1665 DATA CA,9D,9E,F1,B7,20,05,3E          [710]
1666 DATA 07,11,1B,8A,CD,41,9B,CD          [626]
1667 DATA 4B,9B,D2,95,8E,CD,34,9C          [1529]
1668 DATA CD,51,8E,21,10,8A,01,0A          [644]
1669 DATA 00,ED,B0,EB,DD,5E,02,DD          [1064]
1670 DATA 56,03,73,23,72,ED,53,0C          [1535]
1671 DATA 8A,23,DD,5E,00,DD,56,01          [1489]
1672 DATA 73,23,72,ED,53,0E,8A,2A          [1058]
1673 DATA 04,8A,11,0E,00,19,22,0A          [945]
1674 DATA 8A,01,C0,7F,ED,49,CD,92          [1212]
1675 DATA 9B,B7,28,1C,ED,5B,0A,8A          [642]
1676 DATA CD,51,8E,3E,FF,12,2A,04          [1100]
1677 DATA 8A,11,0E,00,19,EB,CD,51          [696]
1678 DATA 8E,EB,CD,F8,9D,C3,9F,8E          [1090]
1679 DATA 36,FF,2A,0A,8A,D5,EB,CD          [1153]
1680 DATA 51,8E,EB,D1,7A,0F,0F,77          [1845]
1681 DATA 01,C0,7F,ED,49,D5,2A,0C          [1539]
1682 DATA 8A,11,1C,A0,01,FF,03,ED          [884]
1683 DATA B0,D1,D5,CD,51,8E,21,1C          [1427]
1684 DATA A0,01,FF,03,ED,B0,01,C0          [1704]
1685 DATA 7F,ED,49,E1,11,FF,03,19          [1257]
1686 DATA 22,0A,8A,2A,0C,8A,19,22          [1180]
1687 DATA 0C,8A,2A,0E,8A,A7,ED,52          [1083]
1688 DATA 22,0E,8A,28,04,38,02,18          [1125]
1689 DATA 8D,ED,5B,0A,8A,CD,51,8E          [450]
1690 DATA EB,36,FF,01,C0,7F,ED,49          [978]
1691 DATA 2A,04,8A,11,0F,00,19,22          [1604]
1692 DATA 04,8A,C9,FE,03,D2,B8,8E          [755]
1693 DATA B7,CA,B8,8E,32,1C,A0,CD          [1161]
1694 DATA 33,9B,DD,6E,00,DD,66,01          [1109]
1695 DATA 22,0C,8A,FE,01,28,04,DD          [1498]
1696 DATA 23,DD,23,CD,34,9C,DD,6E          [856]
1697 DATA 00,DD,66,01,7E,FE,0B,D2          [1688]
1698 DATA 90,8E,23,5E,23,56,B7,20          [1049]
1699 DATA 05,3E,07,11,1B,8A,CD,41          [1660]
1700 DATA 9B,CD,4B,9B,DA,A4,8E,11          [593]
1701 DATA 0E,00,19,7E,32,09,8A,2B          [1187]
1702 DATA 56,2B,5E,ED,53,0E,8A,3A          [1713]
1703 DATA 1C,A0,FE,02,28,08,2B,56          [1654]
1704 DATA 2B,5E,ED,53,0C,8A,01,C0          [1085]
1705 DATA 7F,ED,49,3A,09,8A,4F,CD          [2076]
1706 DATA A6,9B,CD,51,8E,EB,11,1C          [1099]
1707 DATA A0,01,FF,03,ED,B0,7E,32          [1566]
1708 DATA 09,8A,01,C0,7F,ED,49,2A          [873]
1709 DATA 0E,8A,11,FF,03,A7,ED,52          [2389]
1710 DATA 38,20,28,1E,21,1C,A0,ED          [1015]
1711 DATA 5B,0C,8A,01,FF,03,ED,B0          [1445]
1712 DATA ED,53,0C,8A,11,FF,03,2A          [1546]
1713 DATA 0E,8A,A7,ED,52,22,0E,8A          [2074]
1714 DATA 18,B9,21,1C,A0,ED,5B,0C          [1183]
1715 DATA 8A,ED,4B,0E,8A,ED,B0,C9          [424]
1716 DATA 4E,79,06,00,FE,FF,28,0F          [905]
1717 DATA CD,18,9E,CD,A6,9B,CD,51          [1100]
1718 DATA 8E,21,FF,03,19,18,E9,CD          [1471]
1719 DATA 18,9E,01,C0,7F,ED,49,C9          [1689]
1720 DATA C5,ED,5B,06,8A,CD,51,8E          [1086]
1721 DATA EB,C1,09,36,00,C9,FE,01          [1183]
1722 DATA C2,B8,8E,CD,33,9B,CD,34          [875]
1723 DATA 9C,DD,6E,00,DD,66,01,7E          [1246]
1724 DATA FE,0B,D2,90,8E,23,5E,23          [1406]
1725 DATA 56,FE,01,20,07,F5,1A,FE          [685]
1726 DATA 2A,28,52,F1,B7,20,05,3E          [1455]
1727 DATA 07,11,1B,8A,CD,41,9B,CD          [626]
1728 DATA 4B,9B,DA,A4,8E,11,0E,00          [1334]
1729 DATA 19,CD,F8,9D,CD,4B,9B,E5          [1494]
1730 DATA E5,ED,5B,04,8A,CD,51,8E          [1440]
1731 DATA EB,D1,A7,ED,52,11,0F,00          [1183]
1732 DATA ED,52,D1,28,0E,EB,01,0F          [1629]
1733 DATA 00,E5,01,0F,00,09,4B,42          [1585]
1734 DATA D1,ED,B0,01,C0,7F,ED,49          [1797]
1735 DATA 2A,04,8A,11,0F,00,A7,ED          [1669]
1736 DATA 52,22,04,8A,C9,2A,06,8A          [1278]
1737 DATA 11,40,00,19,EB,CD,51,8E          [1401]
1738 DATA EB,3A,08,8A,B7,28,14,5F          [875]
1739 DATA 16,00,A7,ED,52,36,00,3D          [979]
1740 DATA 28,08,4F,06,00,5D,54,13          [977]
1741 DATA ED,B0,23,22,04,8A,01,C0          [1808]
1742 DATA 7F,ED,49,F1,C9,FE,02,C2          [838]
1743 DATA B8,8E,CD,33,9B,CD,34,9C          [611]
1744 DATA DD,6E,02,DD,66,03,7E,FE          [1160]
1745 DATA 0B,D2,90,8E,23,5E,23,56          [572]
1746 DATA FE,01,20,07,F5,1A,FE,2A          [438]
1747 DATA 28,AB,F1,B7,20,05,3E,07          [618]
1748 DATA 11,1B,8A,CD,41,9B,CD,4B          [1681]
1749 DATA 9B,D2,95,8E,01,C0,7F,ED          [1756]
1750 DATA 49,21,10,8A,11,1C,A0,01          [680]
1751 DATA 0A,00,ED,B0,CD,33,9B,DD          [853]
1752 DATA 6E,00,DD,66,01,7E,FE,0B          [1171]
1753 DATA D2,90,8E,23,5E,23,56,B7          [779]
1754 DATA 20,05,3E,07,11,1B,8A,CD          [1602]
1755 DATA 41,9B,CD,4B,9B,DA,A4,8E          [1045]
1756 DATA EB,21,1C,A0,01,0A,00,ED          [1187]
1757 DATA B0,01,C0,7F,ED,49,C9,47          [1690]
1758 DATA 0E,FF,0C,D6,0A,30,FB,79          [1040]
1759 DATA C6,30,FE,30,20,02,3E,20          [1067]
1760 DATA 12,13,79,87,87,81,87,4F          [1490]
1761 DATA 78,91,C6,30,12,C9,B7,C2          [2065]
1762 DATA B8,8E,CD,34,9C,21,22,8A          [1179]
1763 DATA 11,1C,A0,01,17,00,ED,B0          [1318]
1764 DATA ED,5B,06,8A,CD,51,8E,21          [222]
1765 DATA 40,00,19,22,0C,8A,11,33          [964]
1766 DATA A0,3A,05,8A,CB,BF,CB,F7          [1145]
1767 DATA BC,20,06,3A,04,8A,BD,28          [1064]
1768 DATA 36,01,0A,00,ED,B0,3E,20          [948]
1769 DATA 12,13,12,13,D5,23,23,5E          [1490]
1770 DATA 23,56,23,23,7A,E6,FC,0F          [1842]
1771 DATA 0F,3C,D1,CD,47,9F,13,3E          [1608]
1772 DATA 4B,12,06,05,3E,20,13,12          [1219]
1773 DATA 13,10,FC,2A,0C,8A,01,0F          [367]
1774 DATA 00,09,22,0C,8A,18,BA,01          [1021]
1775 DATA C0,7F,ED,49,2A,06,8A,EB          [1016]
1776 DATA CD,51,8E,EB,01,00,40,79          [1201]
1777 DATA BE,20,01,0C,23,10,F9,79          [834]
1778 DATA 01,C0,7F,ED,49,D5,11,39          [1093]
1779 DATA 8A,CD,47,9F,D1,21,36,8A          [955]
1780 DATA 01,0F,00,ED,B0,21,1C,A0          [1426]
1781 DATA C3,BB,8E,B7,C2,B8,8E,CD          [624]
1782 DATA 34,9C,CD,51,8E,EB,11,1C          [1767]
1783 DATA A0,01,00,04,ED,B0,01,C0          [750]
1784 DATA 7F,ED,49,C9,00,00,00,00          [1667]
```

```
10 'RSXDEMO0.BAS                            [1301]
20 'Demoprogramm 1 fuer RSX-Professional    [1857]
30 '(c) 1991 Damir Petkovic & CPC Internat  [2092]
ional
40 |GETMEM,&BF00:var=PEEK(&BF01)*256+PEEK(  [2976]
&BF00)
50 SIR=INT(RND(1)*77)+4:VIS=INT(RND(1)*21)  [3173]
+4
60 IF (sir*vis*8)<var THEN GOSUB 120 ELSE   [3210]
GOTO 80
70 GOTO 40                                  [378]
80 |GETMEM,&BF00:var=PEEK(&BF01)*256+PEEK(  [2976]
&BF00)
90 IF var>65520 THEN GOTO 40                [1121]
100 GOSUB 160                               [933]
110 GOTO 80                                 [306]
```

```
120 GOSUB 180:X=INT(RND(1)*(80-SIR)):Y=INT [3812]
(RND(1)*(25-VIS))
130 okvir=INT(RND(1)*3)+1:col=INT(RND(1)*2 [1844]
)
140 |SETBORDER,col,(col XOR 1),okvir:|OPEN [5448]
WINDOW,X,Y,SIR,VIS,"",""
150 RETURN                                  [555]
160 GOSUB 180:|CLOSEWINDOW                  [1798]
170 RETURN                                  [555]
180 FOR a=0 TO 173:NEXT                     [1133]
190 RETURN                                  [555]
```

```
10 'RSXDEMO1.BAS                            [1253]
20 'Demoprogramm 2 fuer RSX-Professional    [1767]

30 '(c) 1991 Damir Petkovic & CPC Internat [2092]
ional
40 ON BREAK CONT                            [323]
50 BORDER 26:INK 0,26:INK 1,0:INK 2,15:INK [5741]
 3,6:INK 4,20:INK 5,2:INK 6,25:INK 7,3:MOD
E 2:|DEFAULT:|CLEARWINDOWS
60 ' postavljanje tri masinske rutine za p [3636]
romjenu mode-a bez brisanja ekrana
70 FOR X=&AF00 TO &AF0E:READ A:POKE X,A:NE [2087]
XT
80 DATA &3E,0,&C3,&1C,&BD,&3E,1,&C3,&1C,&B [3318]
D,&3E,2,&C3,&1C,&BD
90 ' postavljanje masinskih rutina u inter [3326]
rupt iz Basica !
100 |FASTINTERRUPT:|SETINTERRUPT,&AF0A,&AF [5197]
05,&AF05,&AF00,&AF00,&AF0A
110 ' pocetak demonstracije                 [1190]
120 |SETMODE,2:FOR y=0 TO 4                 [1466]
130 FOR x=0 TO 3:|TEXT,x*17,y,1,0,"RSX Pro [4182]
fessional ":NEXT x
140 NEXT y                                  [359]
150 |SETMODE,1:FOR y=6 TO 17                [2247]
160 FOR x=0 TO 1:|TEXT,x*17,y,1,0,"RSX Pro [3882]
fessional ":NEXT x
170 NEXT y                                  [359]
180 |SETMODE,0:FOR y=20 TO 24               [2063]
190 |TEXT,0,y,1,0,"RSX Professional "       [4042]
200 NEXT y                                  [359]
210 |SETMODE,2:|SETBORDER,1,0,2:|NAME,1,0: [4477]
|SUBNAME,1,0:|INSIDE,0
220 |OPENWINDOW,30,0,20,5," Demonstration  [3949]
"," Press any key "
230 |TEXT,32,2,1,0,"RSX Professional"      [2362]
240 CALL &BB06                              [393]
250 |SETMODE,1:|SETBORDER,0,1,1:|NAME,3,1: [2902]
|SUBNAME,2,1
260 |DELOPTION,1:|DELOPTION,6:|MENUINPUT,& [3651]
BF00,2
270 |MENU,"","Best of Speed Metal","Best o [9234]
f Hard Core","Best of Punk","Best of Class
ics","",9,7," Music List "," Enter your ch
oice "
280 a=PEEK(&BF00)                           [1374]
290 IF a>239 THEN GOTO 450                  [1313]
300 ON (a-1) GOTO 310,360,390,420           [1778]
310 |SETBORDER,1,0,3:|OPENWINDOW,5,a+8,10, [3723]
5,"",""
320 |TEXT,7,a+10,3,0,"Slayer"               [1439]
330 CALL &BB06:|CLOSEWINDOW                 [2303]
340 |MENUONLY,"","Best of Speed Metal","Be [10388]
st of Hard Core","Best of Punk","Best of C
lassics","",9,7," Music List "," Enter you
r choice "
350 GOTO 280                                [413]
360 |SETBORDER,2,0,3:|OPENWINDOW,10,a+8,10 [3493]
,5,"",""
370 |TEXT,12,a+10,1,0,"D.R.I."              [1902]
380 GOTO 330                                [506]
390 |SETBORDER,3,2,3:|OPENWINDOW,15,a+8,17 [3472]
,5,"",""
400 |TEXT,17,a+10,2,0,"Dead Kennedys"       [2552]
410 GOTO 330                                [506]
420 |SETBORDER,2,1,3:|OPENWINDOW,25,a+8,12 [3974]
,5,"",""
430 |TEXT,27,a+10,1,0,"J.S.Bach"            [2204]
440 GOTO 330                                [506]
450 |SETMODE,0:|SETBORDER,4,5,2:|INSIDE,7  [4433]
460 |RESETOPTIONS                           [2097]
```

```
470 |OPENWINDOW,5,20,11,5,"",""             [1556]
480 |TEXT,7,22,6,7,"THE END"                [2114]
490 CALL &BB06:|CLOSEWINDOW:|CLOSEWINDOW:| [3181]
CLOSEWINDOW
500 CALL &BB06:|DEFAULT:MODE 2:END          [2585]
```

```
10 'RSXDEMO2.BAS                            [1779]
20 'Demoprogramm 3 fuer RSX-Professional    [2684]
30 '(c) 1991 Damir Petkovic & CPC Internat [2092]
ional
40 ON BREAK CONT                            [323]
50 MODE 2:PRINT "Wait a minute ..."         [2168]
60 a$="1":b$="6":c$="0"                     [961]
70 RAD                                      [301]
80 DIM x0(162),y0(162),x1(162),y1(162),x2( [3022]
162),y2(162),x3(162),y3(162)
90 c=1                                      [351]
100 FOR a=0 TO 4*PI STEP PI/40              [1550]
110 x0(c)=152+140*SIN(a):y0(c)=178+116*COS [2201]
(a/2)
120 x1(c)=152+128*COS(a/2):y1(c)=178+116*S [2923]
IN(a/2)
130 x2(c)=488+140*SIN(a/2):y2(c)=178+116*C [2008]
OS(a)
140 x3(c)=488+128*COS(a/2):y3(c)=178+116*S [3907]
IN(a)
150 c=c+1:c$=CHR$(VAL(c$)-1+&30):IF c$<"0" [6373]
 THEN c$="9":b$=CHR$(VAL(b$)-1+&30):IF b$<
"0" THEN b$="9":a$=CHR$(VAL(a$)-1+&30)
160 d$=a$+b$+c$:|TEXT,0,2,1,0,d$            [1682]
170 NEXT                                    [350]
180 INK 0,11:INK 1,0:BORDER 11:MODE 2:|DEF [2375]
AULT:|CLEARWINDOWS
190 a=1:b=1:d=1:|RAMDISK,40                 [1591]
200 FOR x=0 TO 639 STEP 20                  [1155]
210 MOVE 0,399:DRAW x,0,1                   [1376]
220 NEXT                                    [350]
230 FOR x=639 TO 0 STEP -20                 [1201]
240 MOVE 639,399:DRAW x,0,1                 [1874]
250 NEXT                                    [350]
260 A$="This  demonstration  was  made"     [3708]
270 B$="to show you how easy you could"     [2736]
280 C$="apply  RSX Professional  using"     [2869]
290 D$="only Basic commands and  still"     [3614]
300 E$="having  an  absolutely  profes-"    [2226]
310 F$="sional  screen display."            [3208]
320 FOR x=1 TO 10:|DELOPTION,x:NEXT         [2628]
330 |SETBORDER,1,0,2:|NAME,1,0              [1823]
340 |MENU,"",a$,b$,c$,d$,e$,f$,""," [7162]
        Author       ","",23,2," Introdu
ction "," Press any key "
350 A$="Now you will be offered"            [3051]
360 B$="to choose  one  of  the"            [2681]
370 C$="options and see some of"            [2735]
380 D$="possibilities  of  this"            [2176]
390 E$="powerfull program."                 [1317]
400 |SETBORDER,0,1,1                        [1819]
410 |MENU,"",a$,b$,c$,d$,e$,"",40,9,""," A [4987]
ny key to procede "
420 |CLOSEWINDOW                            [1184]
430 |TEXT,0,0,1,0,"Metal Bands   RamDisk   [4823]
 Colors   Graphics   System"
440 GOTO 460                                [456]
450 |CLOSEWINDOW                            [1184]
460 |RESETOPTIONS                           [2097]
470 a=1                                     [327]
480 ON d GOTO 490,1320,1740,2170,1990       [623]
490 GOSUB 1030                              [869]
500 |MENU,A$,B$,C$,D$,E$,F$,G$,0,1,"",""    [2440]
510 a=PEEK(&BF00):IF a>239 THEN GOTO 2110   [3371]
520 ON a GOTO 530,590,650,710,770,830,890   [2104]
530 GOSUB 950                               [830]
540 IF b>239 THEN |CLOSEWINDOW:GOSUB 990:G [3502]
OTO 510
550 a$="Tom Araya - Vocals, Bass":b$="Kerr [10877]
y King - Lead Guitar":c$="Jeff Hanneman -
Lead Guitar":d$="Dave Lombardo - Drums":e$
=""
560 f$="Reign In Blood":g$="Angel Of Death [2767]
"
570 GOSUB 1070                              [829]
580 GOTO 540                                [425]
590 GOSUB 950                               [830]
```

```
600 IF b>239 THEN |CLOSEWINDOW:GOSUB 990:G [3502]
OTO 510
610 a$="Max Cavalera - Vocals, Rhythm Guit [11858]
ar":b$="Adreas Kisser - Lead Guitar":c$="P
aulo Jr. - Bass":d$="Igor Cavalera - Drums
":e$=""
620 f$="Beneath The Remains":g$="R.I.P. (R [4970]
est In Pain)"
630 GOSUB 1070                             [829]
640 GOTO 600                               [413]
650 GOSUB 950                              [830]
660 IF b>239 THEN |CLOSEWINDOW:GOSUB 990:G [3502]
OTO 510
670 a$="James Hetfield - Vocals, Lead Guit [9418]
ar":b$="Kirk Hammett - Lead Guitar":c$="Ja
son Newsted - Bass":d$="Lars Ulrich - Drum
s":e$=""
680 f$="Ride The Lightning":g$="Orion"     [3353]
690 GOSUB 1070                             [829]
700 GOTO 660                               [369]
710 GOSUB 950                              [830]
720 IF b>239 THEN |CLOSEWINDOW:GOSUB 990:G [3502]
OTO 510
730 a$="Snake - Vocals":b$="Piggy - Lead G [7663]
uitar":c$="Blacky - Bass":d$="Away - Drums
":e$=""
740 f$="Killing Technology":g$="Tribal Con [4750]
victions"
750 GOSUB 1070                             [829]
760 GOTO 720                               [421]
770 GOSUB 950                              [830]
780 IF b>239 THEN |CLOSEWINDOW:GOSUB 990:G [3502]
OTO 510
790 a$="Joe Belladonna - Vocals":b$="Dan S [10916]
pitz - Lead Guitar":c$="Scott Ian - Rhythm
 Guitar":d$="Frank Bello - Bass":e$="Charl
ie Benante - Drums"
800 f$="Among The Living":g$="Indians"     [2630]
810 GOSUB 1070                             [829]
820 GOTO 780                               [409]
830 GOSUB 950                              [830]
840 IF b>239 THEN |CLOSEWINDOW:GOSUB 990:G [3502]
OTO 510
850 a$="Chuck Billy - Vocals":b$="Alex Sko [11877]
lnick - Lead Guitar":c$="Eric Peterson - R
hythm Guitar":d$="Greg Christian - Bass":e
$="Louie Clemente - Drums"
860 f$="The Legacy":g$="Over The Wall"     [2545]
870 GOSUB 1070                             [829]
880 GOTO 840                               [332]
890 GOSUB 950                              [830]
900 IF b>239 THEN |CLOSEWINDOW:GOSUB 990:G [3502]
OTO 510
910 a$="Steve Grimmett - Vocals":b$="Rob T [11339]
rotman - Lead Guitar":c$="Nige Rockett - L
ead Guitar":d$="James Hinder - Bass":e$="S
teve Grice - Drums"
920 f$="Let There Be Rock":g$="Flame Of Th [5504]
e Antichrist"
930 GOSUB 1070                             [829]
940 GOTO 900                               [320]
950 GOSUB 1220                             [871]
960 |MENU,"",A$,B$,C$,"",6,a+2,"",""       [2500]
970 b=PEEK(&BF00)                          [938]
980 RETURN                                 [555]
990 GOSUB 1030                             [869]
1000 |MENUONLY,A$,B$,C$,D$,E$,F$,G$,0,1,"" [3166]
,""
1010 a=PEEK(&BF00)                         [1374]
1020 RETURN                                [555]
1030 |DEFAULT                              [1268]
1040 A$="Slayer":B$="Sepultura":C$="Metall [7374]
ica":D$="Voivod":E$="Anthrax":F$="Testamen
t":G$="Onslaught"
1050 |SETBORDER,1,0,1:|MENUINPUT,&BF00,INT [4577]
(a)
1060 RETURN                                [555]
1070 |DEFAULT                              [1268]
1080 FOR x=1 TO 10:|DELOPTION,x:NEXT       [2628]
1090 |SETBORDER,1,0,3                      [1470]
1100 ON (b-1) GOTO 1110,1180,1200          [1573]
1110 IF e$="" THEN GOTO 1160               [894]
1120 |MENU,"",a$,b$,c$,d$,e$,"",13,a+b+3," [2924]
",""
1130 |CLOSEWINDOW:GOSUB 1220:|MENUINPUT,&B [3652]
F00,INT(b)
1140 |MENUONLY,"",A$,B$,C$,"",6,a+2,"","" [1885]
1150 b=PEEK(&BF00):GOTO 970                [1478]
1160 |MENU,"",a$,b$,c$,d$,"",13,a+b+3,""," [3414]
"
1170 GOTO 1130                             [345]
1180 |MENU,"",f$,"",13,a+b+3,"",""         [2421]
1190 GOTO 1130                             [345]
1200 |MENU,"",g$,"",13,a+b+3,"",""         [2344]
1210 GOTO 1130                             [345]
1220 |DEFAULT                              [1268]
```

```
1230 A$="Band members":B$="Best LP":C$="Be [4011]
st song"
1240 |DELOPTION,1:|DELOPTION,5             [2857]
1250 RETURN                                [555]
1260 |DEFAULT:|SETBORDER,1,0,3             [2582]
1270 a$="Directory":b$="Load":c$="Save":d$ [3195]
="Rename":e$="Delete"
1280 |MENUINPUT,&BF00,a                    [1544]
1290 |OPENWINDOW,24,21,32,4," Possible Err [2834]
ors ",""
1300 |SETBORDER,1,0,1                      [1472]
1310 WINDOW 26,55,23,24:RETURN             [2786]
1320 ' RamDisk                             [641]
1330 GOSUB 1260:|MENU,a$,b$,c$,d$,e$,14,1, [2161]
"",""
1340 |SETBORDER,0,1,2                      [1822]
1350 a=PEEK(&BF00):IF a>239 THEN |CLOSEWIN [4177]
DOW:GOTO 2110
1360 ON a GOTO 1370,1420,1500,1600,1680    [1800]
1370 |OPENWINDOW,20,a+2,42,17,""," RamDisk [3116]
 Catalogue "
1380 WINDOW #1,22,61,5,19:WINDOW SWAP 0,1  [3211]
1390 |CAT:WINDOW SWAP 0,1                  [2218]
1400 CALL &BB06                            [393]
1410 |CLOSEWINDOW:|MENUONLY,a$,b$,c$,d$,e$ [2881]
,14,1,"","":GOTO 1340
1420 |OPENWINDOW,20,a+2,29,5,""," RamDisk [2978]
Loading "
1430 |TEXT,22,a+4,1,0,"File to load:"      [2130]
1440 WINDOW #2,37,47,a+5,a+5               [1663]
1450 INPUT #2,"",file$                     [685]
1460 |TEXT,22,a+4,1,0,"At  location:"      [1259]
1470 INPUT #2,"",address                   [1066]
1480 IF address=0 THEN |LOAD,file$ ELSE |L [2899]
OAD,file$,address
1490 PRINT:GOTO 1410                       [821]
1500 |OPENWINDOW,20,a+2,29,5,""," RamDisk [1929]
Saving "
1510 WINDOW #2,37,47,a+5,a+5               [1663]
1520 |TEXT,22,a+4,1,0,"File to save:"      [1900]
1530 INPUT #2,"",file$                     [685]
1540 |TEXT,22,a+4,1,0,"From address:"      [1931]
1550 INPUT #2,"",address                   [1066]
1560 |TEXT,22,a+4,1,0,"With  length:"      [1567]
1570 INPUT #2,"",length                    [1144]
1580 |SAVE,file$,address,length            [2810]
1590 PRINT:GOTO 1410                       [821]
1600 |OPENWINDOW,20,a+2,29,5,""," RamDisk [2375]
Renaming "
1610 WINDOW #2,37,47,a+5,a+5               [1663]
1620 |TEXT,22,a+4,1,0,"Old filename:"      [1466]
1630 INPUT #2,"",file1$                    [773]
1640 |TEXT,22,a+4,1,0,"New filename:"      [2551]
1650 INPUT #2,"",file2$                    [779]
1660 |RENAME,file2$,file1$                 [1924]
1670 PRINT:GOTO 1410                       [821]
1680 |OPENWINDOW,20,a+2,29,5,""," RamDisk [3282]
Deleting "
1690 WINDOW #2,37,47,a+5,a+5               [1663]
1700 |TEXT,22,a+4,1,0,"Delete  file:"      [1027]
1710 INPUT #2,"",file$                     [685]
1720 |DELETE,file$                         [1255]
1730 PRINT:GOTO 1410                       [821]
1740 ' Colors                              [632]
1750 |DEFAULT:|MENUINPUT,&BF00,a:|SETBORDE [2876]
R,1,0,1
1760 a$="Pen":b$="Paper":c$="Border"       [2545]
1770 |MENU,a$,b$,c$,24,1,"",""             [2165]
1780 |SETBORDER,0,1,2                      [1822]
1790 a=PEEK(&BF00):IF a>239 THEN GOTO 2110 [3371]
1800 ON a GOTO 1810,1860,1890              [1209]
1810 d$=" new  pen:"                       [225]
1820 GOSUB 1920                            [891]
1830 INK 1,color                           [627]
1840 |MENUONLY,a$,b$,c$,24,1,"",""         [2601]
1850 GOTO 1790                             [341]
1860 d$="new  paper:"                      [353]
1870 GOSUB 1920                            [891]
1880 INK 0,color:GOTO 1840                 [2737]
1890 d$="new border:"                      [928]
1900 GOSUB 1920                            [891]
1910 BORDER color:GOTO 1840                [1517]
1920 |OPENWINDOW,30,a+2,25,5,"",""         [1265]
1930 |TEXT,32,a+4,1,0,"Enter "+d$          [1391]
1940 WINDOW #2,51,53,a+5,a+5               [1573]
1950 INPUT #2,"",color                     [783]
1960 color=ABS(INT(color)):IF color>26 THE [2139]
N GOTO 1950
1970 |CLOSEWINDOW                          [1184]
1980 RETURN                                [555]
1990 ' System                              [749]
2000 |DEFAULT:|MENUINPUT,&BF00,a:|SETBORDE [2876]
R,1,0,1
2010 a$="Return to BASIC":b$="Exit to AMSD [3619]
OS":c$="Exit to CPM"
2020 |MENU,a$,b$,c$,44,1,"",""             [2373]
2030 a=PEEK(&BF00):IF a>239 THEN GOTO 2110 [3371]
```

```
2040 ON a GOTO 2050,2060,2070                     [1364]
2050 |CLOSEWINDOW:|CLOSEWINDOW:|RAMDISKOFF [5940]
:WINDOW 1,80,1,25:PRINT CHR$(10):END
2060 |CLOSEWINDOW:|CLOSEWINDOW:|RAMDISKOFF [3602]
:|BASIC
2070 FOR x=1 TO 10:|DELOPTION,x:NEXT         [2628]
2080 |SETBORDER,1,0,2:|NAME,1,0              [1823]
2090 |MENU,"","MAKE SURE THAT YOUR SYSTEM   [5679]
DISK IS READY !","",17,10," Caution "," Pr
ess any key "
2100 |CLOSEWINDOW:|CLOSEWINDOW:|CLOSEWINDO [4808]
W:|RAMDISKOFF:|CPM
2110 IF a>242 THEN GOTO 2140                 [1148]
2120 d=d-1:IF d=0 THEN d=5                   [1413]
2130 GOTO 450                                [442]
2140 IF a>243 THEN GOTO 450                  [2533]
2150 d=d+1:IF d=6 THEN d=1                   [1776]
2160 GOTO 450                                [442]
2170 ' Graphics                              [576]
2180 |DEFAULT:|MENUINPUT,&BF00,a:|SETBORDE [2876]
R,1,0,1
2190 a$="First graphic":b$="Second graphic [5104]
":c$="Both graphics"
2200 |MENU,a$,b$,c$,33,1,"",""               [2413]
2210 a=PEEK(&BF00):IF a>239 THEN GOTO 2110 [3371]
2220 |SETBORDER,0,1,2                        [1822]
2230 ON a GOTO 2240,2380,2520                [1099]
2240 |OPENWINDOW,42,3,38,18,""," First gra [2778]
phic "
2250 FOR c=1 TO 60                           [762]
2260 MOVE x0(c)+336,y0(c)+32                 [481]
2270 DRAW x1(c)+336,y1(c)+32,1               [392]
2280 NEXT                                    [350]
2290 WHILE INKEY$=""                         [1095]
2300 MOVE x0(c)+336,y0(c)+32                 [481]
2310 DRAW x1(c)+336,y1(c)+32,1               [392]
2320 a=c-60:IF a<0 THEN a=162+a              [1829]
2330 MOVE x0(a)+336,y0(a)+32                 [1070]
2340 DRAW x1(a)+336,y1(a)+32,0               [775]
2350 c=c+1:IF c=163 THEN c=1                 [883]
2360 WEND                                    [390]
2370 GOTO 2730                               [393]
2380 |OPENWINDOW,42,4,38,18,""," Second gr [3045]
aphic "
2390 FOR c=1 TO 60                           [762]
2400 MOVE x2(c),y2(c)+16                     [1140]
2410 DRAW x3(c),y3(c)+16,1                   [1147]
2420 NEXT                                    [350]
2430 WHILE INKEY$=""                         [1095]
2440 MOVE x2(c),y2(c)+16                     [1140]
2450 DRAW x3(c),y3(c)+16,1                   [1147]
2460 a=c-60:IF a<0 THEN a=162+a              [1829]
2470 MOVE x2(a),y2(a)+16                     [783]
2480 DRAW x3(a),y3(a)+16,0                   [578]
2490 c=c+1:IF c=163 THEN c=1                 [883]
2500 WEND                                    [390]
2510 GOTO 2730                               [393]
2520 |OPENWINDOW,0,5,38,18,""," First grap [2731]
hic "
2530 |OPENWINDOW,42,5,38,18,""," Second gr [2578]
aphic "
2540 FOR c=1 TO 60                           [762]
2550 MOVE x0(c),y0(c)                        [582]
2560 DRAW x1(c),y1(c),1                      [170]
2570 MOVE x2(c),y2(c)                        [826]
2580 DRAW x3(c),y3(c),1                      [585]
2590 NEXT                                    [350]
2600 WHILE INKEY$=""                         [1095]
2610 MOVE x0(c),y0(c)                        [582]
2620 DRAW x1(c),y1(c),1                      [170]
2630 MOVE x2(c),y2(c)                        [826]
2640 DRAW x3(c),y3(c),1                      [585]
2650 a=c-60:IF a<0 THEN a=162+a              [1829]
2660 MOVE x0(a),y0(a)                        [575]
2670 DRAW x1(a),y1(a),0                      [498]
2680 MOVE x2(a),y2(a)                        [630]
2690 DRAW x3(a),y3(a),0                      [487]
2700 c=c+1:IF c=163 THEN c=1                 [883]
2710 WEND                                    [390]
2720 |CLOSEWINDOW                            [1184]
2730 |CLOSEWINDOW:|SETBORDER,1,0,1:|MENUON [4257]
LY,a$,b$,c$,33,1,"",""
2740 GOTO 2210                               [361]
```

Elektronikers Freund

Nie wieder Probleme mit Widerständen

Jeder, der gern an elektronischen Schaltungen bastelt, kann wohl ein Lied davon singen: Man hat zwar einen beachtlichen Vorrat an Widerständen, aber der, den man gerade benötigt, ist nicht dabei. Jetzt geht die große Grübelei los, welche Widerstände man am besten kombiniert, um auf den gesuchten Wert zu kommen. Das ist jetzt vorbei.

Das BASIC-Programm "Elektronikers Freund" für den CPC beschäftigt sich mit einem der meistgebrauchten elektronischen Bauteile, dem (Schicht-)Widerstand. Der Umgang mit Widerstands-Werten und -kombinationen ist allerdings nicht immer ganz einfach. Die Widerstände sind bekannterweise nicht mit Dezimalzahlen, sondern mit Farbringen beschriftet. Zu allem Überfluß gibt es auch noch zwei unterschiedliche Codes – das hier abgedruckte Programm verwendet den klassischen Code, der auf drei Ringen für den Widerstandswert beruht. Wenn man nun die Werte der Farben, von denen es immerhin 12 verschiedene gibt, nicht gerade auswendig kennt, bleibt nur der Griff nach einer Umrechnungstabelle ("...aber wo hab' ich sie denn nur hingelegt ?").

Helferlein für Lötkolbenschwinger

Für die Umrechnerei sind die beiden ersten Teile des Programms zuständig. Im Menü erscheinen sie als Punkt eins und zwei. Unter Menüpunkt eins können die Farbringe in eine Dezimalzahl und unter Menüpunkt zwei eine Dezimalzahl in Farbringe umgewandelt werden.

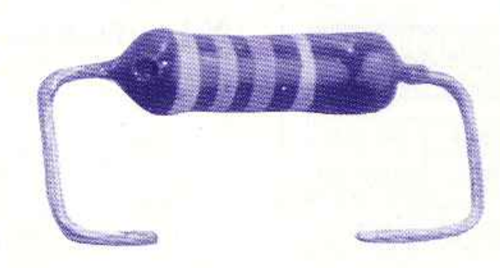
Der dritte Teil des Programms sollte für jeden Elektronikbastler ein besonderer Leckerbissen sein: die Verwaltung der Bestände.

Das hauseigene Sortiment an Widerständen kann noch so gut ausgestattet sein, wenn es drauf ankommt, fehlt unter Garantie der richtige Widerstand.

Da hilft kein Zähneknirschen und Haareraufen, ein Besuch im Elektronikshop ist nun unabdingbar.

Oder etwa doch nicht? Man könnte doch auch zwei Widerstände ersatzweise parallel oder seriell schalten, um den gewünschten Widerstand zu erreichen. Während man die serielle Schaltung von zwei Widerständen noch ganz leicht im Kopf ausrechnen kann (Rg = R1 + R2), vergeht einem schon die Lust, eine Parallelschaltung zu überprüfen (1/Rg = 1/R1 + 1/R2).

Das ist dann genau die richtige Aufgabe für den hauseigenen Rechensklaven – den CPC. Hat man erst einmal seine Widerstände eingegeben (dies kann mittels Dezimalzahl oder auch mittels der Farbringe geschehen), können diese abgespeichert werden. Weiterhin kann der Bestand gelistet und korrigiert werden. Im eigentlich wichtigsten Menüpunkt "Ersatzwiderstand suchen" werden dann alle Widerstände miteinander verglichen und die besten Möglichkeiten als "Top Five" ausgegeben.



1. Umrechnen der Farbringe in Dezimalzahl: Die Auswahl der einzelnen Farben geschieht über das Fenster am rechten Bildschirmrand. Den Zeiger innerhalb dieses Fensters können Sie mittels der Cursortasten hoch und runter bewegen. Mit <RETURN> wählen Sie die Farbe aus.

2. Umrechnen der Dezimalzahl in Farbringe: Zuerst werden Sie nach der Dezimalzahl gefragt, die Sie umwandeln möchten. Sie sollten bei der Eingabe darauf achten, wirklich nur die Zahl einzutippen und keine Leerzeichen vor oder hinter der Zahl, da diese das Ergebnis verfälschen würden. Nachdem die Farben errechnet worden sind, werden Sie gefragt, ob Sie eine Farbdarstellung wünschen. Nach dem Motto: "Ein Bild sagt mehr als tausend Worte", wird der Widerstand mit den errechneten Farbringen darauf dargestellt.

Variablen-Referenz

r(), anz(): in diesen Arrays werden die eingegebenen Widerstände gespeichert.

x1, y1, x2, y2 : Textpositionen für die Umrandung eines Windows für die Unterroutine (Zeile 3910-3970).

i$: dient der Invertierung von Pen- und Paperfarbe.

eing : bei Eins geschah der Aufruf über "Widerstände eingeben", bei Null aus dem Hauptmenü heraus.

zahl : enthält die gewählte Farbe aus dem Farbwahlmenü und wird von der Routine zurückgegeben (Zeile 830).

farbe() : Zwischenspeicherung der gewählten Farben

r : in r wird schrittweise der Widerstand ausgerechnet (Zeile 500-550).

high : Anzahl der Einträge im Farbwahlmenü.

wahly : aktuelle Position im Farbwahlmenü.

wahlyy : alte Position im Farbwahlmenü, was wichtig zum Löschen des alten Zeigers ist (Zeile 770-850).

anzahl : Anzahl der Widerstände plus 1.

dif() : Differenzen zum gewünschten Widerstand. Die geringste Differenz steht am höchsten, deshalb werden die Variablen zu Beginn auch auf 999999 gesetzt (Zeile 1630).

rs : der zu suchende Widerstand (Zeile 1660).

ra : in ra wird jeweils der Widerstand, der sich aus der aktuellen Kombination ergibt, gespeichert (Zeile 1710,1730).

top() : die besten Widerstände aus der "Top Five".

r1(), r2() : die beiden Ausgangswiderstände, die zu einem guten Ergebnis in der "Top Five" führten.

first : ein Flag. Sollte beim Sortieren der "Top Five" keine Veränderung eingetreten sein, wird sie auch nicht neu auf den Bildschirm ausgegeben (Zeile 1930).

flag : Hilfsflag für den Sortieralgorithmus.

proz : die prozentuale Abweichung der Differenz zum gewünschten Widerstand (Zeile 1950,1800).

akty : enthält einen Zeiger innerhalb des r()-Arrays,bei dem mit dem Auflisten begonnen wird (Zeile 3180-3470).

aktend : enthält parallel zu akty einen Zeiger, der auf das letzte Element aus r() zeigt, das gelistet werden soll.

y : enthält die Zeigerposition im Fenster. (Zeile 3300-3470)

oldy : enthält jeweils die alte Zeigerposition im Fenster.

a1,b1, a2,b2 : enthalten die in Grafikkoordinaten umgewandelten Daten aus x1,y1, x2,y2.

3. Ersatzwiderstand suchen: Wenn Sie diesen Menüpunkt wählen, kommen Sie zunächst in ein Untermenü, das neben der eigentlichen Funktion noch weitere nötige Hilfsfunktionen enthält.

Im Menüpunkt *Widerstände eingeben* sollten Sie zunächst Ihre vorhandenen Widerstände eingeben. Dies können Sie entweder mit Hilfe der Farbringe oder mittels Dezimalzahl tun. Die Eingabe mit Hilfe der Farbringe geschieht wie oben beschrieben.

Wenn der Widerstand feststeht, werden Sie weiterhin gefragt, ob Sie nur einen oder mehrere von diesem Widerstand haben. Dies ist wichtig für die Ersatzfunktion.

Die Möglichkeiten innerhalb der *Speichern*-Funktion werden dort weitgehend erläutert. Auch hier wurde versucht, dem Anwender möglichst viel Komfort zukommen zu lassen. Mittels <RETURN> wird einfach ein Standardname gewählt, der auch beim Einladen nach dem selben Prinzip gewählt werden kann.

Bestandspflege

Über den Menüpunkt fünf, *Widerstände auflisten/korrigieren*, kommen Sie schließlich in eine Art Editieroberfläche. In einem Fenster werden die Widerstände aufgelistet.

Mittels <SPACE> können Sie weiterblättern. Mit den Cursortasten (rauf und runter) kann ein Widerstand angewählt werden.

In dem kleinen Fenster rechts können Sie nun wieder über die Cursortasten (rechts und links) auswählen. Die Bestätigung erfolgt wieder über <RETURN>.

Das Programm funktioniert auf allen CPC-Typen. Tippen Sie es ab und speichern Sie es beispielsweise unter dem Namen EF.BAS.

Nun, jedem Elektroniker der öfter mit Widerständen jongliert, ist geholfen.

Selbst aus dem größten Sortiment sollte es problemlos möglich sein, eine passende Widerstandskombination herauszufinden. Die Suchroutine benötigt zwar etwas Zeit, aber dies sollte in einem solchen Fall keine Rolle spielen. Die manuelle Suche kann auf alle Fälle noch länger dauern. Also Schluß mit den Problemen mit unpassenden Werten.

Christian Leubner/jg

```
10 '**********************************    [1460]
20 '  ELEKTRONIKERS       F R E U N D      [837]
30 '                                       [117]
40 '  (c)1991 Christian Leubner            [1097]
45 '   & CPC International                 [2118]
50 '**********************************    [1460]
60 DIM zahl(20),s$(20)                     [462]
70 DIM r(100),anz(100)                     [272]
80 MODE 2                                  [513]
90 CALL &BC02:PEN 1:PAPER 0                [1067]
100 INK 1,26:BORDER 0:INK 0,0              [1402]
110 WINDOW 2,79,6,24                       [961]
120 x1=1:y1=5:x2=80:y2=25                  [1947]
130 GOSUB 3910                             [1003]
140 WINDOW #2,2,79,2,3                     [1094]
150 x1=1:y1=1:x2=80:y2=4                   [1516]
160 GOSUB 3910                             [1003]
170 CLS                                    [91]
180 CLS#2                                  [372]
190 PRINT#2,SPC(28)"- Elektronikers Freund [4042]
 -"
200 PRINT#2,SPC(22)"written by Christian L [4400]
eubner in 1991"
210 i$=CHR$(24)                            [1139]
220 LOCATE 34,2:PRINT i$" Hauptmenue "i$   [1538]
230 LOCATE 24,5:PRINT i$"[1]"i$"  Farbring [4971]
e in Zahl umrechnen"
240 LOCATE 24,7:PRINT i$"[2]"i$"  Zahl in  [6582]
Farbringe umrechnen"
250 LOCATE 24,9:PRINT i$"[3]"i$"  Ersatzwi [4267]
derstand heraussuchen"
260 LOCATE 30,12:PRINT i$" Bitte waehlen : [3591]
 "i$;
270 a$=INKEY$:IF a$="" THEN 270            [1338]
280 a=VAL(a$)                              [956]
290 IF a<1 OR a>3 THEN 270                 [907]
300 PRINT a                                [401]
310 ON a GOSUB 360,930,1450                [1069]
320 GOTO 170                               [427]
330 '*****                                 [174]
340 'Umrechnung: Farbringe in Zahlen       [2957]
350 '*****                                 [174]
360 CLS                                    [91]
370 WINDOW #3,67,77,8,21                   [1275]
380 x1=66:y1=7:x2=78:y2=22                 [1095]
390 GOSUB 3910                             [1003]
400 LOCATE 1,1:PRINT"Bitte geben Sie die F [4679]
arbringe von links nach rechts ein:"
410 FOR z=1 TO 4                           [1043]
420 IF z=4 AND eing=1 THEN 490             [1378]
430 GOSUB 620                              [921]
440 IF zahl=-1 THEN r=-1:RETURN            [865]
450 farbe(z)=zahl                          [415]
460 LOCATE 1,3+z                           [837]
470 IF z=4 THEN PRINT                      [773]
480 PRINT " "i$"Farbring";z;":"i$" ";s$    [3138]
490 NEXT z                                 [354]
500 r=10*farbe(1)                          [886]
510 r=r+farbe(2)                           [1296]
```

```
520 IF farbe(3)<1 AND farbe(3)>0 THEN 550  [905]
530 r=r*(10^farbe(3))                      [1290]
540 GOTO 560                               [373]
550 r=r*farbe(3)                           [789]
560 IF eing=1 THEN RETURN                  [1547]
570 LOCATE 1,12                            [724]
580 PRINT"Der Widerstand betraegt"r;CHR$(1 [4450]
91)
590 PRINT"   mit einer Toleranz von"farbe( [2442]
4)"%"
600 PRINT:PRINT"-Taste-":CALL &BB06        [1598]
610 RETURN                                 [555]
620 RESTORE 870                            [725]
630 IF z=3 THEN high=12 ELSE high=10       [2081]
640 IF z=4 THEN RESTORE 890:high=4         [696]
650 GOSUB 680                              [972]
660 RETURN                                 [555]
670 '****                                  [104]
680 'Auswahlfenster: Farben                [1689]
690 '****                                  [104]
700 CLS#3                                  [371]
710 FOR i=1 TO high                        [866]
720 READ s$(i),zahl(i)                     [742]
730 PRINT#3,s$(i):NEXT                     [1075]
740 high=high+1                            [403]
750 s$(high)="ABBRUCH":zahl(high)=-1       [2102]
760 PRINT#3,s$(high)                       [573]
770 wahly=1:wahlyy=wahly                   [1102]
780 LOCATE#3,11,wahlyy:PRINT#3," ":LOCATE# [3223]
3,11,wahly:PRINT#3,"<"
790 wahlyy=wahly                           [404]
800 a$=INKEY$:IF a$="" THEN 800            [1304]
810 IF a$=CHR$(240) THEN wahly=wahly-1     [702]
820 IF a$=CHR$(241) THEN wahly=wahly+1     [992]
830 IF a$=CHR$(13) THEN s$=s$(wahly):zahl= [2837]
zahl(wahly):RETURN
840 IF wahly<1 THEN wahly=1                [925]
850 IF wahly>high THEN wahly=high          [1397]
860 GOTO 780                               [409]
870 DATA schwarz,0,braun,1,rot,2,orange,3, [5593]
gelb,4,gruen,5,blau,6
880 DATA violett,7,grau,8,weiss,9,gold,0.1 [1969]
,silber,0.01
890 DATA braun,1,rot,2,gold,5,silber,10    [1674]
900 '*****                                 [174]
910 'Umrechnung: Zahl in Farbringe         [1620]
920 '*****                                 [174]
930 CLS                                    [91]
940 INPUT"Gesuchter Widerstand: ",a$       [3365]
950 IF a$="" THEN RETURN                   [1147]
960 farbe(3)=LEN(a$)-2                     [1140]
970 IF INSTR(1,a$,".")>0 THEN 1100         [627]
980 farbe(1)=VAL(LEFT$(a$,1))              [944]
990 farbe(2)=VAL(MID$(a$,2,1))             [1862]
1000 RESTORE 870                           [725]
1010 FOR i=0 TO 11                         [508]
1020 READ s$(i),zahl(i)                    [742]
1030 NEXT                                  [350]
1040 FOR i=1 TO 3                          [435]
```

```
1050 LOCATE 1,4+i:PRINT"Farbring"i": "s$(f [3323]
arbe(i))
1060 NEXT i                                   [375]
1070 PRINT:PRINT"Moechten Sie eine Farbdar [7078]
stellung ? [ /n]"
1080 a$=UPPER$(INKEY$):IF a$="" THEN 1080  [4246]
ELSE IF a$="N" THEN RETURN ELSE 1180
1090 GOTO 1180                                [399]
1100 zahl=VAL(a$)                            [1469]
1110 IF LEN(a$)=3 THEN farbe(3)=10:zahl=za [3577]
hl*10
1120 IF LEN(a$)=4 THEN farbe(3)=11:zahl=za [3009]
hl*100
1130 a$=RIGHT$(STR$(zahl),LEN(STR$(zahl))- [2070]
1)
1140 GOTO 980                                 [496]
1150 '****                                    [104]
1160 'Farbdarstellung des Widerstandes       [2755]
1170 '****                                    [104]
1180 MODE 1                                   [506]
1190 INK 0,10:BORDER 10                       [980]
1200 a$=CHR$(206)                             [515]
1210 PRINT TAB(5);STRING$(31,a$)             [1111]
1220 PRINT TAB(4);STRING$(33,a$)             [1098]
1230 PRINT TAB(4);STRING$(33,a$)             [1098]
1240 PRINT TAB(3);STRING$(35,a$)             [1671]
1250 PRINT STRING$(39,a$)                    [1502]
1260 PRINT TAB(3);STRING$(35,a$)             [1671]
1270 PRINT TAB(4);STRING$(33,a$)             [1098]
1280 PRINT TAB(4);STRING$(33,a$)             [1098]
1290 PRINT TAB(5);STRING$(31,a$)             [1111]
1300 RESTORE 1410                             [749]
1310 FOR i=0 TO 11:READ zahl(i):NEXT         [1128]
1320 FOR i=1 TO 3                             [435]
1330   FOR j=1 TO 9                           [604]
1340      IF farbe(i)>9 THEN a$=CHR$(207) E [2847]
LSE a$=CHR$(143)
1350      LOCATE i*10,j:INK i,zahl(farbe(i) [4249]
):PEN i:PRINT STRING$(3,a$)
1360   NEXT j                                 [370]
1370      LOCATE i*10,j+2:PRINT s$(farbe(i) [2616]
)
1380 NEXT i                                   [375]
1390 CALL &BB06                               [393]
1400 RUN                                      [243]
1410 DATA 0,3,6,15,24,9,1,5,13,26,24,26      [1670]
1420 '*****                                   [174]
1430 'Untermenue: Ersatzwiderstand herauss [2907]
uchen
1440 '*****                                   [174]
1450 anzahl=1                                 [179]
1460 CLS                                       [91]
1470 LOCATE 34,2:PRINT i$" Untermenue "i$   [2135]
1480 LOCATE 24,5:PRINT i$"[1]"i$"  Ersatzw [3692]
iderstand suchen"
1490 LOCATE 24,7:PRINT i$"[2]"i$"  Widerst [5766]
aende eingeben"
1500 LOCATE 24,9:PRINT i$"[3]"i$"  Widerst [3914]
aende abspeichern"
1510 LOCATE 24,11:PRINT i$"[4]"i$"  Widers [3179]
taende einladen"
1520 LOCATE 24,13:PRINT i$"[5]"i$"  Bestan [3562]
d listen/korrigieren"
1530 LOCATE 24,15:PRINT i$"[6]"i$"  zuruec [4312]
k zum Hauptmenue"
1540 LOCATE 30,18:PRINT i$" Bitte waehlen  [2421]
: "i$;
1550 a$=INKEY$:IF a$="" THEN 1550            [1435]
1560 a=VAL(a$):IF a<1 OR a>6 THEN 1550       [1197]
1570 PRINT a                                  [401]
1580 IF a=6 THEN RETURN                      [1027]
1590 ON a GOTO 1630,2100,2740,2930,3090      [1714]
1600 '****                                    [104]
1610 'Untermenue: Ersatzwiderstand suchen    [2733]
1620 '****                                    [104]
1630 FOR i=1 TO 6:dif(i)=999999:NEXT         [1219]
1640 CLS                                       [91]
1650 PRINT i$" Bestmoeglichen Ersatzwiders [4259]
tand suchen "i$
1660 PRINT:INPUT"Welchen Widerstand moecht [4174]
en Sie erreichen: ",rs
1670 IF rs=0 THEN 1460                       [1101]
1680 FOR i=1 TO anzahl-1                     [1241]
1690  IF anz(i)<2 THEN x=i+1 ELSE x=i        [1782]
1700   FOR j=x TO anzahl-1                   [1654]
1710     ra=r(i)+r(j):a$="seriell"           [1752]
1720     GOSUB 1830                           [901]
1730     ra=(1/r(i))+(1/r(j)):ra=ROUND(1/r [4010]
a):a$="parallel"
1740     GOSUB 1830                           [901]
1750   NEXT j                                 [370]
1760 NEXT i                                   [375]
1770 LOCATE 1,12                              [724]
1780 PRINT" Die beste Kombination, die ich [5746]
 anbieten kann, ist:"
1790 PRINT r(r1(1))CHR$(191)" und"r(r2(1)) [4317]
CHR$(191)" "s$(1)" zu schalten."
1800 proz=ROUND((dif(1)*100)/rs)             [1587]
1810 PRINT" Die Abweichung vom gewuenschte [6269]
n Widerstand betraegt"proz"%"
1820 PRINT:PRINT"-Taste-":CALL &BB06:GOTO  [2179]
1460
1830 '***                                      [53]
1840 'Top Five ausgeben                      [1101]
1850 '***                                      [53]
1860 top(6)=ra:s$(6)=a$:r1(6)=i:r2(6)=j:di [3532]
f(6)=ABS(ra-rs)
1870 first=0                                  [237]
1880 flag=0                                   [263]
1890 FOR k=1 TO 5                            [1081]
1900    IF dif(k)>dif(k+1) THEN GOSUB 2000    [618]
1910 NEXT k                                   [373]
1920 IF flag=1 THEN first=1:GOTO 1880        [1643]
1930 IF first=0 THEN RETURN                  [1148]
1940 FOR k=1 TO 5                            [1081]
1950 proz=ROUND((dif(k)*100)/rs)             [2064]
1960 LOCATE 1,k+4                             [907]
1970 PRINT"-"k"- R"top(k)CHR$(191)" (Abw.: [6391]
"dif(k)" --"proz"%) / aus"r(r1(k))CHR$(191
)" &"r(r2(k))CHR$(191)" "s$(k);STRING$(15,
32)
1980 NEXT                                     [350]
1990 RETURN                                   [555]
2000 help=top(k):top(k)=top(k+1):top(k+1)= [2343]
help
2010 help$=s$(k):s$(k)=s$(k+1):s$(k+1)=hel [1823]
p$
2020 help=r1(k):r1(k)=r1(k+1):r1(k+1)=help [1423]
2030 help=r2(k):r2(k)=r2(k+1):r2(k+1)=help  [943]
2040 help=dif(k):dif(k)=dif(k+1):dif(k+1)= [1118]
help
2050 flag=1                                   [262]
2060 RETURN                                   [555]
2070 '****                                    [104]
2080 'Untermenue: Widerstaende eingeben      [2308]
2090 '****                                    [104]
2100 CLS                                       [91]
2110 WINDOW #5,4,32,12,12                    [1410]
2120 x1=3:y1=11:x2=33:y2=13                  [1464]
2130 GOSUB 3880                               [977]
2140 LOCATE 2,3:PRINT"Wie moechten Sie Ihr [4918]
e Widerstaende eingeben ?"
2150 '***                                      [53]
2160 'Auswahlfenster: Art der Eingabe waeh [1839]
len
2170 '***                                      [53]
2180 x=1                                      [350]
2190 CLS#5:IF x=1 THEN 2200 ELSE IF x=2 TH [2716]
EN 2220
2200 PRINT#5," "I$" als Zahl "i$"   als Fa [2972]
rbringe"
2210 GOTO 2230                                [309]
2220 PRINT#5,"  als Zahl   "i$" als Farbri [2983]
nge "i$
2230 a$=INKEY$:IF a$="" THEN 2230            [1546]
2240 IF a$=CHR$(243) OR a$=CHR$(242) THEN  [1477]
2270
2250 IF a$=CHR$(13) THEN GOTO 2280           [1402]
2260 GOTO 2230                                [309]
2270 x=(x XOR 3):GOTO 2190                   [1315]
2280 IF x=1 THEN 2320 ELSE 2430               [969]
2290 '***                                      [53]
2300 'Eingabeart 1: als Zahlen               [2126]
2310 '***                                      [53]
2320 CLS                                       [91]
2330 PRINT i$" Eingabe der Widerstaende al [3125]
s Zahlen "i$
2340 PRINT                                    [361]
2350 INPUT"Widerstand: ",r(anzahl)           [3495]
2360 IF r(anzahl)=0 THEN 2520                [1710]
2370 GOSUB 2540                               [879]
2380 anzahl=anzahl+1                          [722]
2390 GOTO 2320                                [347]
2400 '***                                      [53]
2410 'Eingabeart 2: als Farbringe            [2316]
2420 '***                                      [53]
2430 CLS:eing=1:GOSUB 360:eing=0             [1716]
2440 IF r=-1 THEN 2520                        [953]
2450 r(anzahl)=r                              [733]
2460 CLS                                       [91]
2470 PRINT i$"Eingabe der Widerstaende mit [5072]
 Hilfe der Farbringe"i$
2480 PRINT:PRINT"Der eingegebene Widerstan [5239]
d betraegt"r;CHR$(191)
```

```
2490 GOSUB 2540                                    [879]
2500 anzahl=anzahl+1                               [722]
2510 GOTO 2430                                     [349]
2520 PRINT:PRINT"Moechten Sie die eingegeb [6739]
enen Widerstaende abspeichern ? [j/n]"
2530 a$=UPPER$(INKEY$):IF a$="J" THEN 2740 [3433]
 ELSE IF a$="N" THEN 1460 ELSE 2530
2540 WINDOW #4,4,27,12,13                          [1489]
2550 x1=3:y1=11:x2=28:y2=14                        [1682]
2560 GOSUB 3880                                    [977]
2570 CLS#4                                         [386]
2580 PRINT#4,"Wieviele Widerstaende ?"             [2611]
2590 x=1                                           [350]
2600 IF x=1 THEN 2610 ELSE 2630                    [735]
2610 LOCATE#4,1,2:PRINT#4," "i$" einen "i$ [3205]
"  mehrere  "
2620 GOTO 2640                                     [363]
2630 LOCATE#4,1,2:PRINT#4,"  einen   "I$"  [3433]
mehrere "i$" "
2640 a$=INKEY$                                     [278]
2650 IF a$=CHR$(242) OR a$=CHR$(243) THEN  [787]
2680
2660 IF a$=CHR$(13) THEN 2690                      [730]
2670 GOTO 2640                                     [363]
2680 x=(x XOR 3):GOTO 2600                         [1155]
2690 IF x=1 THEN anz(anzahl)=1 ELSE anz(an [2388]
zahl)=2
2700 RETURN                                        [555]
2710 '****                                         [104]
2720 'Untermenue: Widerstaende abspeichern [2840]
2730 '****                                         [104]
2740 CLS                                           [91]
2750 IF anzahl=0 THEN PRINT i$" Keine Wide [4231]
rstaende eingegeben "i$:CALL &BB06:GOTO 14
60
2760 PRINT i$"Eingegebene Widerstaende abs [4021]
peichern"i$
2770 PRINT:PRINT"Mit [Return] wird der Nam [3608]
e >R-DATA.DAT< gewaehlt."
2780 PRINT"Mit [.] und [Return] koennen Si [4358]
e abbrechen."
2790 PRINT:INPUT"Dateiname: ",a$                   [1178]
2800 IF a$="." THEN 1460                           [926]
2810 IF a$="" THEN a$="r-data.dat"                 [1355]
2820 OPENOUT a$                                    [765]
2830 WRITE#9,anzahl                                [1307]
2840 FOR i=1 TO anzahl                             [1073]
2850 WRITE#9,r(i),anz(i)                           [1566]
2860 NEXT:CLOSEOUT                                 [904]
2870 PRINT:PRINT i$" Widerstaende abgespei [3342]
chert ! "i$
2880 CALL &BB06                                    [393]
2890 GOTO 1460                                     [391]
2900 '***                                          [53]
2910 'Untermenue: Widerstaende einladen            [1815]
2920 '***                                          [53]
2930 CLS                                           [91]
2940 PRINT i$" Widerstaende einladen "i$           [3299]
2950 PRINT:PRINT"Mit [Return] wird der Nam [3608]
e >R-DATA.DAT< gewaehlt."
2960 PRINT"Mit [.] und [Return] koennen Si [4358]
e abbrechen."
2970 PRINT:INPUT"Dateiname: ",a$                   [1178]
2980 IF a$="." THEN 1460                           [926]
2990 IF a$="" THEN a$="r-data.dat"                 [1355]
3000 OPENIN a$                                     [715]
3010 INPUT#9,anzahl                                [934]
3020 FOR i=1 TO anzahl                             [1073]
3030 INPUT#9,r(i),anz(i)                           [584]
3040 NEXT:CLOSEIN                                  [753]
3050 GOTO 1460                                     [391]
3060 '****                                         [104]
3070 'Untermenue: Widerstaende auflisten/k [2648]
orrigieren
3080 '****                                         [104]
3090 CLS                                           [91]
3100 PRINT i$" Widerstaende auflisten/korr [4355]
igieren "i$
3110 PRINT:PRINT i$" [Return] "i$" zum Bes [3160]
tand korrigieren."
3120 PRINT i$"  [Space] "i$" um weiterzuli [1901]
sten"
3130 PRINT i$"      [S] "i$" um die Widers [6336]
taende nach ihrem Wert zu sortieren"
3140 PRINT i$"      [.] "i$" um abzubreche [2425]
n"
3150 WINDOW #3,4,44,23,13                          [1200]
3160 x1=3:y1=12:x2=45:y2=24                        [1989]
3170 GOSUB 3880                                    [977]
3180 akty=1                                        [391]
3190 CLS#3                                         [371]
3200 aktend=akty+9                                 [566]
```

```
3210 IF aktend>=anzahl THEN aktend=anzahl- [1859]
1
3220 IF anzahl=1 THEN CLS:PRINT i$" Keine  [9101]
Widerstaende (mehr) vorhanden ! "i$:CALL &
BB06:GOTO 1460
3230 FOR i=akty TO aktend                          [1304]
3240 IF anz(i)=1 THEN a$="einer" ELSE a$=" [2595]
mehrere"
3250 PRINT#3,"R"r(i);CHR$(191)," - Bestand [2803]
: "a$
3260 NEXT i                                        [375]
3270 '***                                          [53]
3280 'Auswahlfenster: Liste der Widerstaen [1844]
de
3290 '***                                          [53]
3300 y=1:oldy=y                                    [685]
3310 LOCATE#3,39,oldy:PRINT#3," ":LOCATE#3 [2043]
,39,y:PRINT#3,"<"
3320 oldy=y                                        [525]
3330 a$=INKEY$:IF a$="" THEN 3330                  [1431]
3340 IF a$=CHR$(240) THEN y=y-1:GOTO 3410          [1243]
3350 IF a$=CHR$(241) THEN y=y+1:GOTO 3410          [1630]
3360 IF a$=" " THEN 3450                           [659]
3370 IF a$=CHR$(13) THEN 3510                      [782]
3380 IF a$="." THEN 1460                           [926]
3390 IF a$="s" OR a$="S" THEN GOTO 3790            [2352]
3400 GOTO 3330                                     [463]
3410 IF y<1 THEN y=1                               [770]
3420 IF y>10 THEN y=10                             [900]
3430 IF y+akty>anzahl THEN y=anzahl-akty           [3073]
3440 GOTO 3310                                     [451]
3450 IF akty+10>anzahl THEN akty=1:GOTO 31 [3308]
90
3460 akty=akty+10                                  [503]
3470 GOTO 3190                                     [634]
3480 '***                                          [53]
3490 'Auswahlfenster: Bestimmung der vorha [2738]
ndenen Stueckzahl
3500 '***                                          [53]
3510 WINDOW #4,47,77,15,15                         [1488]
3520 x1=46:y1=14:x2=78:y2=16                       [1783]
3530 GOSUB 3880                                    [977]
3540 PRINT#4,"Anzahl :"                            [1421]
3550 ON x GOTO 3560,3570,3580                      [1135]
3560 LOCATE#4,1,2:PRINT#4," "i$" einer "i$ [5224]
"  mehrere    keiner  ":GOTO 3590
3570 LOCATE#4,1,2:PRINT#4,"  einer   "i$"  [5122]
mehrere "i$"  keiner  ":GOTO 3590
3580 LOCATE#4,1,2:PRINT#4,"  einer    mehr [3766]
ere   "i$" keiner "i$" "
3590 a$=INKEY$:IF a$="" THEN 3590                  [1474]
3600 IF a$=CHR$(242) THEN x=x-1:GOTO 3640          [1156]
3610 IF a$=CHR$(243) THEN x=x+1:GOTO 3640          [1036]
3620 IF a$=CHR$(13) THEN 3670                      [742]
3630 GOTO 3590                                     [491]
3640 IF x<1 THEN x=1                               [512]
3650 IF x>3 THEN x=3                               [1027]
3660 GOTO 3550                                     [467]
3670 IF x=1 THEN anz(akty+y-1)=1                   [1919]
3680 IF x=2 THEN anz(akty+y-1)=2                   [1790]
3690 IF x=3 THEN 3700 ELSE 3190                    [440]
3700 beg=akty+y-1:ende=anzahl-1                    [1891]
3710 FOR i=beg TO ende                             [908]
3720 r(i)=r(i+1):anz(i)=anz(i+1)                   [1367]
3730 NEXT:r(anzahl-1)=0:anz(anzahl-1)=0            [3278]
3740 anzahl=anzahl-1                               [734]
3750 GOTO 3190                                     [634]
3760 '***                                          [53]
3770 'Untermenue Listen/Korrigieren: Wider [4226]
staende nach Wert sortieren
3780 '***                                          [53]
3790 flag=0                                        [263]
3800 FOR i=1 TO anzahl-2                           [1248]
3810   IF r(i)>r(i+1) THEN GOSUB 3840              [1002]
3820 NEXT                                          [350]
3830 IF flag=1 THEN 3790 ELSE 3180                 [706]
3840 help=r(i):r(i)=r(i+1):r(i+1)=help             [1458]
3850 help=anz(i):anz(i)=anz(i+1):anz(i+1)= [2957]
help
3860 flag=1                                        [262]
3870 RETURN                                        [555]
3880 '****                                         [104]
3890 'Windowumrahmung zeichnen                     [1820]
3900 '****                                         [104]
3910 a1=8*x1-2:b1=400-(16*y1-10)                   [1310]
3920 a2=8*x2-6:b2=400-(16*y2-4)                    [2287]
3930 MOVE a1,b1:DRAW a2,b1:DRAW a2,b2:DRAW [2888]
 a1,b2:DRAW a1,b1
3940 a1=a1-4:a2=a2+4                               [966]
3950 b1=b1-8:b2=b2+8                               [393]
3960 MOVE a1,b1:DRAW a2,b1:DRAW a2,b2:DRAW [2888]
 a1,b2:DRAW a1,b1
3970 RETURN                                        [555]
```

Frisch kopiert und editiert...

Zwei alltagstaugliche Arbeitshilfen als Zugaben auf der neuen CPC-DATABOX

Nach dem Motto "Nicht nur das Allerneueste kann gefallen" haben wir in Programmen gestöbert, die schon ein paar Donnerstage auf dem Buckel haben. Dabei sind wir auf zwei Anwendungen gestoßen, die auch heute noch hochaktuell sind und in die Schublade "praktische Helfer" gehören.

Wer die "CPC International" kennt, für den ist auch die Sache mit dem "Bonus" nichts Neues: Es geht dabei um Programme, die wir aus Platzgründen zwar nicht im Heft veröffentlichen können, unseren Lesern aber trotzdem nicht vorenthalten wollen. Darum packen wir sie als Zugaben auf die DATABOX, unsere Service-Diskette zum Heft. Die beiden "Bonüsse" dieses Heftes wurden bereits 1987 entwickelt und laufen auf allen CPCs. Sie stammen aus einer hochkarätigen, umfangreichen Programmsammlung, die wir angekauft haben. Die besten Stücke daraus wollen wir in unregelmäßigen Abständen unseren Lesern auf Diskette verfügbar machen.

Der "Mal-eben"-Editor

Ein Texteditor ist wie eine Packung Taschentücher: Man sollte ihn immer zur Hand haben. Wollen Sie Textdateien für eigene Adventure-Spiele herstellen? Möglicherweise schreiben Sie auch Programme in C, Pascal oder Assembler, oder Sie bekommen gelegentlich Disketten mit Anwendungen, deren Anleitung als ASCII-Datei abgespeichert ist: In all diesen Fällen kann Ihnen "Super-Edit 1.1" weiterhelfen. Er wird vom BASIC aus gestartet, braucht nur 4 kByte im Speicher und stellt die wichtigsten Funktionen zur Bearbeitung von Texten zur Verfügung. Das Vorwärts- und Rückwärtsrollen durch einen Text geschieht dabei wesentlich schneller, als Sie es von den meisten Editorprogrammen gewohnt sein dürften.

Neben den üblichen Tastenfunktionen, die man vom System des CPC her kennt, gibt es noch einige Sonderkommandos, die sich mit den Funktionstasten aufrufen lassen:

Mit <f1> geht man im Text um 20 Zeilen weiter, während <f4> um 20 Zeilen zurückführt.

Der Textanfang läßt sich mit <f5>, das Textende mit <f2> anspringen.

Die Umschaltung zwischen Überschreib- und Einfügemodus geschieht mit <f6>. Zum Löschen eines Wortes wird <f9> betätigt. Mit <f7> geht man ein Wort nach links, mit <f8> eines nach rechts.

<f3> schließlich dient zum Einfügen von Makros. Das sind vordefinierte Floskeln oder Textpassagen, die jeweils höchstens 50 Zeichen lang sein dürfen. Bis zu 9 Makros lassen sich definieren. Zum Einfügen drückt man zuerst <f3>, dann je nach gewünschtem Makro <f1> bis <f9>.

<f0> führt zum Menü, wo sich unter anderem der Befehl zum Definieren der Makros befindet. Einige Funktionen des <f0>-Menüs sind es wert, kurz erklärt zu werden:

<F> setzt den Dateinamen fürs Laden oder Abspeichern von Texten. Solange hier nichts angegeben ist, kann man weder laden noch speichern.

<A>,<B> und <S> dienen zum Suchen und/oder Ersetzen von Wörtern. Wenn das Programm nach Drücken von <S> im Text ein Wort gefunden hat, das "Wort A" entspricht, kann man entweder weitersuchen lassen (<S>), die Suche abbrechen (<E>), zurück in die Texteingabe gehen (<Return>) oder das Wort an der angezeigten Stelle durch "Wort B" ersetzen (C).

<I> fügt den Inhalt einer Textdatei in den augenblicklich bearbeiteten Text ein und beginnt damit an der aktuellen Cursorposition.

<M> erlaubt die Eingabe von Textpassagen für die Makros. Das jeweils zu definierende Makro wird aus der angezeigten Liste ausgewählt.

<O> speichert die Makroliste auf Diskette, <P> lädt eine entsprechende Liste von dort.

<C> löscht nach einer Sicherheitsabfrage den gesamten Text im Speicher.

Erfreulicherweise unterstützt "Super-Edit 1.1" deutsche Umlaute, ß und Paragraphenzeichen. Die Tastenbelegung:

Ä/a = eckige/geschweifte Klammer auf
Ö/ö = Backslash/senkrechter Strich
Ü/ü = eckige/geschweifte Klammer zu
ß = <Control>-<2>
§ = A-Round ("Klammeraffe")

Die deutschen "Spezialitäten" wurden so in den ASCII-Code hineindefiniert, daß sie sich mit praktisch allen gängigen Textdateien vertragen.

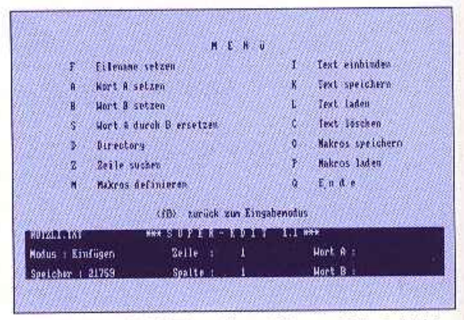
Der Sicherheitskopierer

Das zweite Bonus-Programm stellt 1:1-Sicherheitskopien von Disketten her. Es arbeitet wahlweise mit einem Laufwerk oder mit zweien. Das Kopieren geht spurweise vor sich; jeweils sieben Spuren werden von der Quelldiskette in den Speicher gelesen, analysiert, und dann auf die Zieldiskette geschrieben. Ein bestehender Kopierschutz wird dabei mit übertragen. Das klappt natürlich nicht bei allen Programmen. Bei "Winter Games", "Way Of The Tiger" und "Builderland" zum Beispiel war unser Programm aber durchaus erfolgreich.

Es gibt Kopierschutztechniken, die es erfordern, als Zieldiskette eine ganz neue, unformatierte Diskette zu verwenden. Dies ist jedoch nicht in jedem Fall nötig.

Übrigens: Klar, daß unser Track-Kopierer nur für Sicherheitskopien zum persönlichen Archivier-Bedarf gedacht ist. Als Werkzeug für professionelle Raubkopierer eignet er sich ohnehin nicht, weil er vom kopiergeschützten Original wieder ein geschütztes Duplikat erzeugt und den Schutz nicht etwa "knackt".

(sz)



Der flinke ASCII-Texteditor "Super-Edit 1.1" bietet seine wichtigsten Funktionen per Menü an

Schöne Schleifchen

Pascal-Kolleg: Schleifenprogrammierung in Turbo Pascal und Basic

Auch diese Folge des Pascal-Kollegs bietet wieder allen Joyce- und CPC-Programmierern, die mit Turbo Pascal arbeiten, Grundlagenwissen rund um Code und Struktur. Diesmal geht es um Schleifenkonstruktionen und das nötige Drumherum.

In praktisch allen höheren Programmiersprachen kennt man Zählschleifen. Ihr Zweck besteht darin, daß ein bestimmter Anweisungsblock mehrfach abgearbeitet wird und die Anzahl der Durchläufe von vornherein feststeht.
In FORTRAN gibt es hierzu zum Beispiel die *DO*-Anweisung, in LOGO den *REPEAT*-Befehl und in BASIC die *FOR..TO*-Konstruktion mit der Option *STEP*. Etwas Ähnliches kennt natürlich auch Pascal: Es gibt die *FOR..TO..DO* -Schleifen (für wachsenden Laufindex) und *FOR..DOWNTO..DO*-Schleifen (für fallenden Laufindex).
Jedoch ist ein wichtiger Unterschied zu beachten. In Pascal darf der Laufindex nicht vom Typ *REAL* sein, während das in BASIC die Regel ist. In Pascal kommen nur die sogenannten Ordinaltypen in Frage, wie zum Beispiel die ganzen Zahlen (Integer) und die Zeichen (Char).
Das folgende BASIC-Programm dient dem Ziel, eine Tabelle der Quadrate aller Zahlen zwischen eins und drei mit der Schrittweite 0.05 zu erzeugen:

```
10 FOR X=1 TO 3 STEP 0.05
20 PRINT X,X×X
30 NEXT X
```

Die Variable X steht in diesem BASIC-Programm unzweifelhaft für eine REAL-Zahl. Wie können wir jedoch das gegebene Programm in Pascal übersetzen? Versuchen wir zunächst einmal eine Konstruktion mit eine Zählerschleife! Benötigt wird ein Laufindex, den wir als *INTEGER*-Variable L wählen.
In welchen Grenzen soll *L* variieren? Dazu müssen wir berechnen, wieviele Werte benötigt werden. Von $X=1$ bis $X=2$ sind es zwanzig, von $X=2$ bis $X=3$ noch einmal zwanzig; zusammen macht das ... 41, denn den letzten Wert drei dürfen wir nicht vergessen. Es ist daher sinnvoll, den Wert *L* von eins bis 41 laufen zu lassen. Als nächstes muß aus dem jeweiligen Wert von *L* (der als solcher niemanden ernsthaft interessiert) der eigentlich gemeinte Real-Wert *X* ermittelt werden. Dazu stellen wir eine kleine Tabelle auf:

```
für L=1 wollen wir X=1,
für L=2 wollen wir X=1.05,
für L=3 wollen wir X=1.10,
für L=4 wollen wir X=1.15,
...
```

REAL – Schleifen in PASCAL

Ein bißchen Nachdenken führt zu dem Ergebnis, daß der Wert der Hauptvariablen *X* aus *L* allgemein wie folgt berechnet werden kann:

```
X:=1+(L-1)×0.05
```

Damit sind wir schließlich in der Lage, den entsprechenden Pascal-Programmteil zu schreiben:

```
Var L:Integer; X:Real;
...
For L:=1 To 41 Do
Begin
X:=1+(L-1)×0.05
WriteLn(X:10:4, X×X:10:4);
End;
```

Dieses Programmstück ist korrekt, jedoch erscheint die Umrechnung von *L* in *X* kompliziert.

Aufgabe 1: Es ist besser, die Schleife von $L:=0$ bis 40 laufen zu lassen. Formulieren Sie das entsprechende Programmstück!

Auf jeden Fall lauern in der mentalen Arithmetik Fehlerquellen, auch wenn man sich das Leben durch geeignete Wahl der Grenzen vereinfacht. Und ist es nicht ärgerlich, für einen solch simplen Vorgang einige Minuten Denkarbeit investieren zu müssen?
Glücklicherweise bietet die Sprache Pascal weitere Konstruktionen zur Erzeugung von Schleifen: Es gibt noch die *REPEAT..UNTIL*-Schleifen und die *WHILE..DO*-Schleifen. Exemplarisch werden wir die eben als *FOR..TO*-Schleife realisierte Konstruktion nun als *REPEAT..UNTIL*-Schleife formulieren. Die Laufvariable *L* wird hier nicht mehr benötigt. Die Real-Variable *X* wird anfangs auf den Startwert $X=1$ gesetzt und danach so lange in Schritten von 0.05 heraufgezählt, bis der Endwert überschritten ist.

```
Var X:Real; {Vorsicht!}
...
X:=1;
Repeat
WriteLn(X:10:4,X×X:10:4);
X:=X+0.05;
Until X:=3.05;
```

Der Vorteil einer solchen Konstruktion ist – neben dem Einsparen der Variablen für der Laufindex – das Vermeiden der umständlichen Umrechnung von der Laufvariablen *L* in die Hauptvariable *X*.
Allerdings liegt hier noch ein Haar in der Suppe: In der angegebenen Konstruktion ist ein Fehler verborgen, der sich erst zur Laufzeit bemerkbar macht.

Aufgabe 2: Worin liegt der Fehler und wie kann er behoben werden?

Wolfgang J. Weber/
Andreas Knösel/rs

```
BASIC
FOR <Var> = <Wert1> TO <Wert2> STEP <Schrittweite> :NEXT <Var>
WHILE <Bedingung=Wahr> WEND
PASCAL
FOR <Var1> = <Wert1> TO <Wert2> DO <Ausführung>
REPEAT <Ausführung> UNTIL <Bedingung=Wahr>
WHILE <Bedingung=True> DO <Ausführung>
```

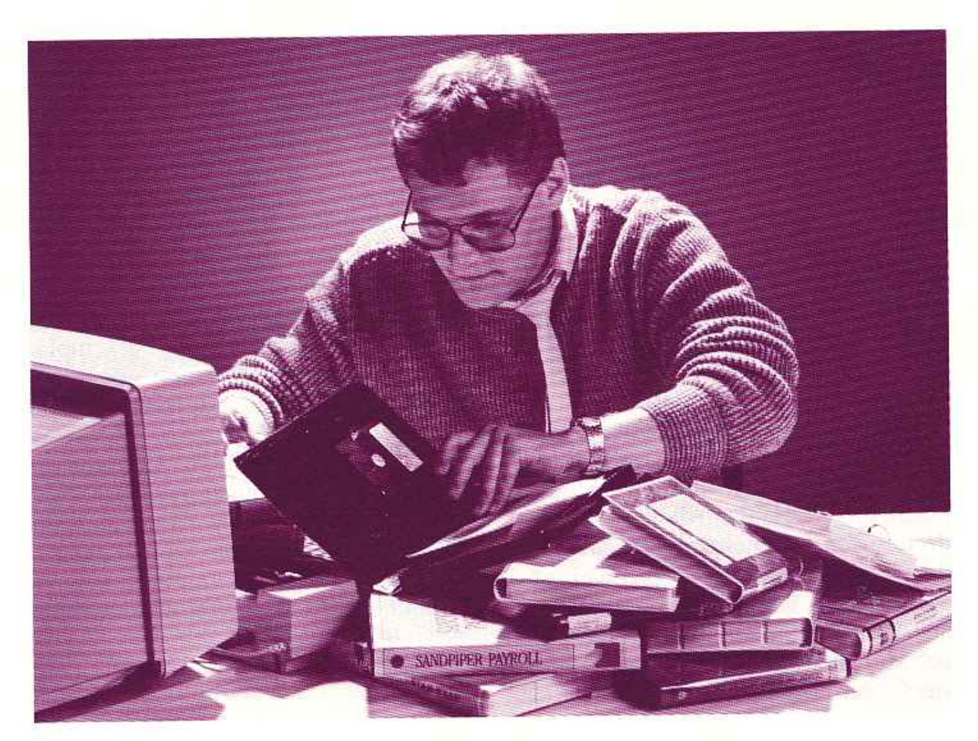
Alle Schleifenfunktionen in Pascal und Basic auf einen Blick

```
For L: = 1 To 40 Do
Begin
X: = 1 + L × 0.05;
WriteLn(X:10:4,X × X:10:4);
End;
```

Lösung zu Aufgabe 1: Wenn die Schleife von L:=0 bis 40 läuft, wird die Umrechnungsformel von L in X einfacher: X:=1+L×0.05

```
...
Repeat
WriteLn(X:10:4,XX:10:4);
X: = X + 0.05
Until X > 3.02;
```

Lösung zu Aufgabe 2: Die angegebene Schleife endet nicht! Der Grund ist, daß hier für die Abbruchbedingungen zwei Real-Zahlen auf Gleichheit überprüft werden, bei denen aber Rundungsfehler auftreten können. Als Regel sollte man beherzigen: Real-Zahlen sollten stets nur auf Ungleichheit getestet werden.



Alles in Ordnung?

Sortierverfahren unter der Lupe

Wer einen Computer sein Eigen nennt, wird sicherlich auch bald anfangen, auf diesem eigene Programme zu entwickeln. Gerade Dateiverarbeitungen erfreuen sich bei Einsteigern sehr großer Beliebtheit. Aber was tun, wenn die Daten unsortiert in irgendwelchen Feldern stehen?

Sollte man eine Dateiverarbeitung oder irgend etwas ähnliches entwickelt haben, wo viele Daten in ein- oder mehrdimensionalen Feldern verwaltet werden, kann es immer recht nützlich sein, diese in einer ordentlichen Form nach bestimmten Kriterien soriert vorliegen zu haben.
Eine Sortierroutine muß also her.

Die Qual der Wahl

Nun ist es aber so, daß es in der Informatik für jedes Problem mehrere Lösungen gibt. Da sich schon vor vielen Jahren Wissenschaftler dieser Problematik angenommen haben, fallen diese Lösungen logischerweise immer anders aus. Die einen – die auf kurze und logische Ergebniss aus waren – entwickelten Routinen mit nur wenigen Programmzeilen. Die anderen, die mehr auf einen Geschwindigkeitsvorteil als auf ein kurzes Programm aus waren, entwickelten kompliziertere Verfahren wie zum Beispiel Quick-Sort mit seinen rekursiven Aufrufen.

Quick-Sort

Dieses Verfahren wurde von dem amerikanischen Mathematiker C.A.R. Hoare entwickelt. Hierbei wird ein vorgegebenes Feld so in zwei Teile aufgeteilt, daß in einem Teilfeld die Elemente stehen, die kleiner als ein gewisses "trennendes" Element sind, im anderen Teil die größeren. Anschließend werden beide Teilfelder nach dem gleichen Muster behandelt, und dies solange, bis man bei einelementigen Mengen angelangt ist. Nun ist das Feld sortiert. Als Beispiel nehmen wir einmal die Zahlenfolge 98 89 07 85 32 23 72 78 38 13. Bitte schauen Sie sich für die weitere Erklärung die Abbildung "Das Quicksort-Verfahren" an. Als trennendes Element wird das jeweils in der Mitte stehende genommen (durch () gekennzeichnet).

Es wird jeweils, zunächst von links beginnend, das erste Element gesucht, das größer als das oder gleich dem Trennelement ist; dann von rechts her das erste Element, welches kleiner oder gleich dem Trennelement ist. Diese beiden Elemente werden vertauscht.

Select-Sort

Im Gegensatz hierzu basiert das Select-Sort-Verfahren (oder auch Sortieren durch Auswahl genannt) auf einem wesentlich einfacheren Prinzip. In dem Datenfeld A wird zunächst das kleinste Element herausgesucht und an die vorderste Position gesetzt, das bedeutet, das Element a[1] und das minimale Element tauschen die Plätze (wird im Programm von der Prozedur Switch erledigt).
Unter den verbliebenen Elementen a[2], a[3],...a[n] wird nun wieder das kleinste Element gesucht und mit a[2] getauscht. Dies geht nun immer so weiter, bis das vorletzte Datenelement erreicht wird.

Heap-Sort

Dieser Algorithmus, den der Amerikaner J. Williams 1964 vorstellte, ist die Weiterentwicklung des Select-Sort-Verfahrens. Hier wird in einem Feld mit der Länge n das erste Element zur Wurzel eines Baumes erklärt. Jedes Element dieses Feldes ist zugleich ein Knoten des Binärbaumes, dem wieder ein linker und ein rechter Nachfolger zugeordnet werden kann. Ein zehn Elemente großes Feld sieht dann zum Beispiel so aus:
1
2 3
4 5 6 7
8 9 10
Auf dieser Struktur basierend kann nun ein Heap (Haufen) aufgebaut werden. Ein Feld heißt Heap, wenn für alle n Elemente des Feldes folgendes gilt:
Der Wert des i-ten Feldelementes ist größer oder gleich dem Wert des 2xi-ten und des 2xi +1-ten Feldelemnts
(i, 2xi, 2xi+1 sind kleiner oder gleich n)
Diese Bedingung erzwingt einen Binärbaum, in dem der 'Vater' größer oder gleich seinem linken und rechten Sohn ist.

Verfahren	Minuten:Sekunden
Quick-Sort	0:31
Heap-Sort	0:36
Shell-Sort	0:38
Select-Sort	1:49
Insert-Sort	1:51
Bubble-Sort	2:48

Solange benötigen die Sortier-Algorithmen, um tausend Datensätze zu sortieren.

Das größte Element eines Heaps befindet sich stets an der ersten Stelle des Feldes. Ein solches Heap wird auch Maximal-Heap genannt. Analog zu dieser Form kann ein Minimal-Heap definiert werden. Für dieses Heap gilt dann die Regel, daß der "Vater" nicht größer sein darf, als sein rechter oder linker "Sohn".
Bei der eigentlichen Sortierung, kann man also in zwei Schritte unterteilen:

- Das Feld wird zu einem Heap transformiert
- Das erste (**kleinste** oder größere) Element wird mit dem letzten (**größten** oder kleinsten) Element vertauscht und das Feld um das letzte Feldelement verkürzt.

Werden diese zwei Schritte entsprechend der Anzahl der Felder minus eins mal wiederholt, ist das Feld sortiert.
Eine Lösung dieser Problematik können Sie dem Listing Heap.Inc entnehmen.

Insert-Sort

besser Sortieren durch Einfügen ist eine weitere interessante Art, Daten in die richtige Ordnung zu bringen. Die Geschwindigkeit ist ungefähr mit der beim Select-Sorting zu vergleichen.
Bei der Entwicklung dieses Verfahrens ging man davon aus, daß ein vorsortiertes Feld vorliegt, in welches ein weiteres Element aufgenommen werden soll. Das jeweils neu hinzukommende Element wird nun –bei Stelle 1 beginnend – mit den Listenelementen verglichen. Es wandert nun solange zu größeren Platznummern, bis es auf ein Listenelement stößt, daß größer als es selbst ist. Damit ist die Einfügestelle gefunden. Jetzt werden alle größeren Listenelemente um eine Stelle nach rechts geschoben und das Element wird an der gefundenen Stelle eingefügt.

Shell-Sort

Das Shell-Sort-Verfahren (1959, D.L. Shell) ist wiederum eine Weiterentwicklung eines bereits vorhandenen Algorithmus. Das Sortieren durch Einfügen wird hier noch effizienter realisiert, da hier Verschiebeoperationen über größere Entfernungen hinweg durchgeführt werden. Im ersten Durchlauf wird das Datenfeld in zwei Teilfolgen mit der Länge des Gesamtfeldes Div 2 sortiert. Logischerweise beträgt nun auch die Entfernung der Elemente der Teilfolgen untereinander genau die Länge des Gesamtfeldes Div 2. Nun werden alle Teilfolgen nacheinander sortiert. Im zweiten Durchlauf wird die Entfernung zwischen den Elementen einer Teilfolge halbiert; die Länge der Teilfolgen verdoppelt sich dann automatisch. Hiernach werden diese neuen Teilfolgen nochmals sortiert. Auch in den nächsten Durchläufen wird dieses Verfahren wiederholt, und im letzen Durchgang beträgt die Entfernung zwischen den Elementen einer Folge eins, was bedeutet, daß nur eine Teilfolge vorliegt, deren Länge der Feldlänge entspricht. Im Vergleich zum Insert-Sort-Verfahren erzielt Shell-Sort wirklich ein enorm gutes Ergebnis.

Bubble-Sort

Ein einfaches, jedoch nicht gerade schnelles Verfahren stellt die Austauschroutine da. Hierbei wandern die Werte der zwei Schleifenvariablen (i,j) jeweils von zwei bis zur Datensatzgröße (i) und von der Datensatzgröße bis i (j). Die zweite For-Schleife übernimmt hier das Durchlaufen des Restfeldes mit den impliziten Vergleichen und den entsprechenden Verschiebungen. Die äußere For-Schleife übernimmt das kontinuierliche Verkürzen des Restfeldes.

Beispielprogramme

Die als Listing ausgedruckten Beispielprogramme stellen die einzelnen Sortierverfahren noch einmal als Pascal-Listing da. Das Hauptprogramm SORT.PAS bindet die einzelnen Programmteile ein und demonstriert die einzelnen Sortierverfahren. Es ist grundsätzlich auch dafür zu gebrauchen, um nach dem Eingeben erst einmal die einzelnen Sortierverfahren auf richtiges Eingeben zu überprüfen.
Nach dem Start von SORT.PAS wird jeweils das LeerArray mit durcheinandergewürfelten Integerwerten gefüllt. Die Inhalte des unsortierten Arrays werden nun auf dem Bildschirm ausgegeben und im Speicher mit dem jeweiligen Verfahren sortiert. Nun findet eine Ausgabe des sortierten Arrays statt, und es wird mit dem nächsten Sortierverfahren weitergemacht.
Um eine Kompatibilität des Programms auch mit anderen CP/M-Maschinen zu gewährleisten, wurde der Geschwindigkeitstest unter CP/M ausgeführt. Hierzu wurde ein Programm geschrieben, daß so aussah:

```
Program Test;
Type StudentArray = Array[1..1000] Of In-
teger;
     Student = Integer;
Var TestArray: StudentArray;
    Loop: Integer;
{$IBubble.Inc}
Procedure Fill;
Begin
 For Loop:=1 to 1000 Do
  TestArray[Loop]:=Random(1000)+1;
End;
Begin { Test }
Fill;
BubbleSort(TestArray,1000);
End. { Test }
```

Die hier fett markierten Punkte sollten nun je nach dem gewünschten Sortierverfahren entsprechend abgeändert werden.
Der eigentliche Test kann nun direkt unter CP/M durch Eingabe von

```
DATE!TEST!DATE
```

gestartet werden.
Voraussetzung hierfür ist jedoch, daß sich das CP/M-Dienstprogramm DATE.COM auf dem aktuellen Laufwerk befindet.
Subtrahiert man nun die erste angegebene Zeit von der zweiten, kommt man auf die benötige Zeit für das Sortieren. Hierbei werden zwar auch Diskettenzugriffe mit einbezogen, sie sind jedoch bei jedem Verfahren gleich und fallen somit nicht sonderlich ins Gewicht.
Außerdem bietet sich auch die Möglichkeit an, wenn man im Besitz einer RAM-Disk ist, alle benötigten Dateien auf diese zu kopieren.

Ralf Schößler

Empfohlene Literatur:

- Pascal, Samuel L. Marateck, John Wiley & Sons, ISBN 0-471-60546-8
- Pascal at Work and Play, Richard S. Forsyth, Capman and Hall, ISBN 0-412-23380-0
- Das Arbeitsbuch zu Turbo-Pascal, Karl Udo Bromm, Sybex, ISBN 3-88745-629-7
- Programmieren mit Pascal, Rüdeger Baumann, Vogel Verlag, ISBN 3-8023-0667-8
- Programmieren mit Turbo-Pascal 3.0, Winfried Kassera, Markt & Technik, ISBN 3-89090-159-X

```
13 89 07 85(32)23 72 78 38 98
13 23 07 85(32)89 72 78 38 98
13 23 07(32)85 89 72 78 38 98
13 07(23)32 85 89 72 78 38 98
07(13)23 32 85 89 72 78 38 98
07 13 23 32 38 89(72)78 85 98
...
```

Das Shell-Sort-Verfahren

```
{ Programm: Sort.Pas                                  }
{ Funktion: Laedt die einzelnen Module ein und        }
{           sortiert das zuvor belegte TestArray      }
{ Autor: Ralf Schoessler-Niebergall                   }
{ (c) CPC International                               }

Program SortTester;

Const Size = 100;    { Groesse des TestArrays }

Type StudentArray = Array [1..Size] Of Integer;
     Student      = Integer; { Art der Daten }

Var TestArray:   StudentArray;
    Loop:        Integer; { Schleifenvariable }

{$IBubble.Inc}           { Einbinden der einzelnen }
{$ISelect.Inc}           { Sortierverfahren        }
{$IInsert.Inc}
{$IHeap.Inc}
{$IQuick.Inc}
{$IShell.Inc}

Procedure Fill;         { Fuellt das TestArray mit }
Begin                   { Integer-Werten }
 For Loop:=1 To Size do
  TestArray[Loop]:=Random(Size)+1;
 WriteLn('Shuffling');
End;

Procedure OutPut; { Ausgabe der Werte auf Bild- }
Begin             { schirm }
 For Loop:=1 to Size Do
  Begin
   Write (TestArray[Loop]:6,' : ');
  End;
 WriteLn;
End;

Begin                                  { SortTester }

 ClrScr; Fill; OutPut;          { BubbleSort }
 WriteLn('BubbleSorting');
 BubbleSort(TestArray,Size);
 WriteLn('Ready'); OutPut;

 ClrScr; Fill; OutPut;          { SelectSort }
 WriteLn('SelectSorting');
 SelectSort(TestArray,Size);
 WriteLn('Ready'); OutPut;

 ClrScr; Fill; OutPut;          { InsertSort }
 WriteLn('InsertSorting');
 InsertSort(TestArray,Size);
 WriteLn('Ready'); OutPut;

 ClrScr; Fill; OutPut;            { HeapSort }
 WriteLn('HeapSorting');
 HeapSort(TestArray,Size);
 WriteLn('Ready'); OutPut;

 ClrScr; Fill; OutPut;           { QuickSort }
 WriteLn('QuickSorting');
 QuickSort(TestArray,Size);
 WriteLn('Ready'); OutPut;

 ClrScr; Fill; OutPut;           { ShellSort }
 WriteLn('ShellSorting');
 ShellSort(TestArray,Size);
 WriteLn('Ready'); OutPut;

 Repeat Until KeyPressed;
End.

{ Ende Sort.Pas                                       }
```

```
{ Bubble.Inc                                          }
{ Sortierverfahren: Bubble-Sort                       }
{ Uebergabeparameter:                                 }
{         Class (das Array vom Typ StudentArray       }
{         ClassSize (Letzte Datensatznummer           }
{ Vordefiniert muss sein:                             }
{  StudentArray (TYPE StudentArray= Array Of ...)     }
{  Student (Datentyp)                                 }
{ Aufruf:                                             }
{ BubbleSort (DatenArray, letzter Datensatz)          }

Procedure BubbleSort   (Var Class: StudentArray;
                        ClassSize: Integer);
Var i,j: Integer; { Schleifenvariablen }

 Procedure Switch (Var Stu1, Stu2: Student);
 { Vertauscht Inhalt von zwei Variablen }
 Var TempStu: Student; { Austauschvariable }
 Begin                                   { Switch }
  TempStu:=Stu1;
  Stu1:=Stu2;
  Stu2:=TempStu;
 End;                                    { Switch }

Begin                                { BubbleSort }
 For i:=2 To ClassSize Do
 For j:=ClassSize DownTo i Do
  If Class[j-1] > Class[j] Then
   Switch (Class[j-1],Class[J]);
End;                                 { BubbleSort }

{ Ende Bubble.Inc                                     }
```

```
{ Select.Inc                                          }
{ Sortierverfahren: Select-Sort                       }
{ Uebergabeparameter:                                 }
{         Class (das Array vom Typ StudentArray)      }
{         ClassSize (Letzte Datensatznummer)          }
{ Vordefiniert muss sein:                             }
{  StudentArry (TYPE StudentArray= Array Of ... )     }
{  Student (Datentyp)                                 }
{ Aufruf:                                             }
{ SelectSort (DatenArray, letzter Datensatz)          }

Procedure SelectSort (Var Class: StudentArray;
                      ClassSize: Integer);

Var i, j,                  {Schleifenvariablen}
    IndexOfMax: Integer;   {Groesster Wert}

 Procedure Switch (Var Stu1, Stu2 : Student);
 { Vertauscht Inhalt von zwei Variablen }
 Var TempStu: Student; { Austauschvariable }
 Begin                                   {Switch}
  TempStu:=Stu1; Stu1:=Stu2; Stu2:=TempStu;
 End;                                    {Switch}

Begin                                {SelectSort}
 For i:=ClassSize DownTo 2 Do
 Begin
  IndexOfMax:=i;
  For j:=i-1 DownTo 1 Do
   If Class[j] > Class[IndexOfMax] Then
   IndexOfMax:=j;
  If IndexOfMax<>i Then
  Switch (Class[i], Class[IndexOfMax]);
 End;
End;                                 {SelectSort}

{ Ende Select.Inc                                     }
```

```
{ Insert.Inc                                          }
{ Sortierverfahren: Insert-Sort                       }
{ Uebergabeparameter:                                 }
{         Class (das Array vom Typ StudentArray       }
{         ClassSize (Letzte Datensatznummer)          }
{ Vordefiniert muss sein:                             }
{  StudentArray (TYPE StudentArray= Array Of ...)     }
{  Student (Datentyp)                                 }
{ Aufruf:                                             }
{ InsertSort (DatenArray, letzter Datensatz)          }

Procedure InsertSort (Var Class: StudentArray;
                      ClassSize: Integer);

Var i,j,k: Integer; { Schleifenvariablen }
    h:     Student; { Austauschvariable }

Begin                                { InsertSort }
 For i:=2 To ClassSize Do
 Begin
  h:=Class[i]; j:=1;
  While (h>Class[j]) And (j<i) Do j:=j+1;
  If j<i Then For k:=i DownTo j+1 Do Class[k]:=
                                      Class[k-1];
  Class[j]:=h; { Class[j] einfuegen }
 End;
End;                                 { InsertSort }

{ Ende Insert.Inc                                     }
```

```
{ Heap.Inc                                                    }
{ Sortierverfahren: Heap-Sort                                 }
{ Uebergabeparameter:                                         }
{        Class (das Array vom Typ StudentArray)               }
{        ClassSize (Letzte Datensatznummer)                   }
{ Vordefiniert muss sein:                                     }
{  StudentArray (TYPE StudentArray= Array Of ...)             }
{  Student (Datentyp)                                         }
{ Aufruf:                                                     }
{ HeapSort (DatenArray, letzter Datensatz                     }

Procedure HeapSort (Var Class: StudentArray;
                    ClassSize: Integer);

 Procedure Switch (var Stu1, Stu2: Integer);
 { Vertauscht Inhalt von zwei Variablen }
 Var TempStu: Student; { Austauschvariable }
 Begin                                          { Switch }
  TempStu:=Stu1; Stu1:=Stu2; Stu2:=TempStu;
 End;                                           { Switch }

 Procedure filter_down (max_ebene: Integer);
 Var i,                       { Schleifenvariable }
     Child     : Integer;
     ready     : Boolean;
 Begin                                     { filter_down }
  i:=1; ready:=false;
  While (Not ready) Do
  Begin
   Child:=2 * i;
   If (Child > max_ebene) Then
   ready:=True
   Else
   Begin
    If (Child+1 <= max_ebene) then
     If (Class[Child+1] > Class[Child]) Then
     Child:=Child+1;
    If (Class[i] < Class[Child]) Then
    Begin
     Switch(Class[i], Class[Child]);
     i:=Child;
    End
    Else
     ready:=True;
   End;
  End;
 End;                                      { filter_down }

 Procedure filter_up (max_ebene: Integer);
 Var i,                 { Schleifenvariable }
     parents: Integer;
 Begin                                       { filter_up }
  i:=max_ebene;
  While (i <> 1) Do
  Begin
   parents:=i Div 2;
   If (Class[i] > Class[parents]) Then
   Begin
    Switch(Class[parents], Class[i]);
    i:=parents;
   End
   Else
   i:=1;
  End;
 End;                                        { filter_up }

Var i: Integer;  { Schleifenvariable }

Begin                                        { Heap_Sort }
 For i:=2 TO ClassSize Do filter_up(i);
 For i:=ClassSize DownTo 2 Do
 Begin
  Switch (Class[1], Class[i]);
  filter_down (i-1);
 End;
End;                                         { Heap_Sort }

{ Ende Heap.Inc                                               }
```

```
{ Quick.Inc                                                   }
{ Sortierverfahren: Quick-Sort                                }
{ Uebergabeparameter:                                         }
{        Class (das Array vom Typ StudentArray                }
{        ClassSize (Letzte Datensatznummer                    }
{ Vordefiniert muss sein:                                     }
{  StudentArray (TYPE StudentArray= Array Of ...)             }
{  Student (Datentyp)                                         }
{ Aufruf:                                                     }
{ QuickSort (DatenArray, letzter Datensatz)                   }

{$a-} { Rekursiven Code erzeugen!!! }

Procedure QuickSort (Var Class: StudentArray;
                     ClassSize: Integer);

 Procedure Switch (Var Stu1, Stu2: Student);
 { Vertauscht Inhalt von zwei Variablen }
 Var TempStu: Student; { Austauschvariable }
 Begin                                          { Switch }
  TempStu:=Stu1;
  Stu1:=Stu2;
  Stu2:=TempStu;
 End;                                           { Switch }

 Procedure QSort (Klein, Gross: Integer);
 Var i, j:  Integer; { Schleifenvariable }
     m:     Student;
 Begin                                           { QSort }
  i:=Klein;
  j:=Gross;
  m:=Class[j];
  Repeat
   While Class[i] < m Do i:=i+1;{ Class[i] > m ? }
   While Class[j] > m Do j:=j-1;{ Class[j] > m ? }
   If i<=j Then
   Begin
    Switch (Class[j],Class[i]); { austauschen }
    i:=i+1; j:=j-1;      { i u. j annaehern }
   End;
  Until i>j;
  If Klein<j Then QSort(Klein,j); { Rekursion }
  If Gross>i Then QSort(i,Gross); { Rekursion }
 End;                                            { QSort }

Begin                                        { QuickSort }
 QSort(1,ClassSize);
End;                                         { QuickSort }

{$a+} { Rekursion ausschalten }

{ Ende Quick.Inc                                              }
```

```
{ Shell.Inc                                                   }
{ Sortierverfahren: Shell-Sort                                }
{ Uebergabeparameter:                                         }
{        Class (das Array vom Typ StudentArray                }
{        ClassSize (Letzte Datensatznummer)                   }
{ Vordefiniert muss sein:                                     }
{  StudentArray (TYPE StudentArray= Array Of...)              }
{  Student (Datentyp)                                         }
{ Aufruf:                                                     }
{ ShellSort (DatenArray, letzter DatenSatz                    }

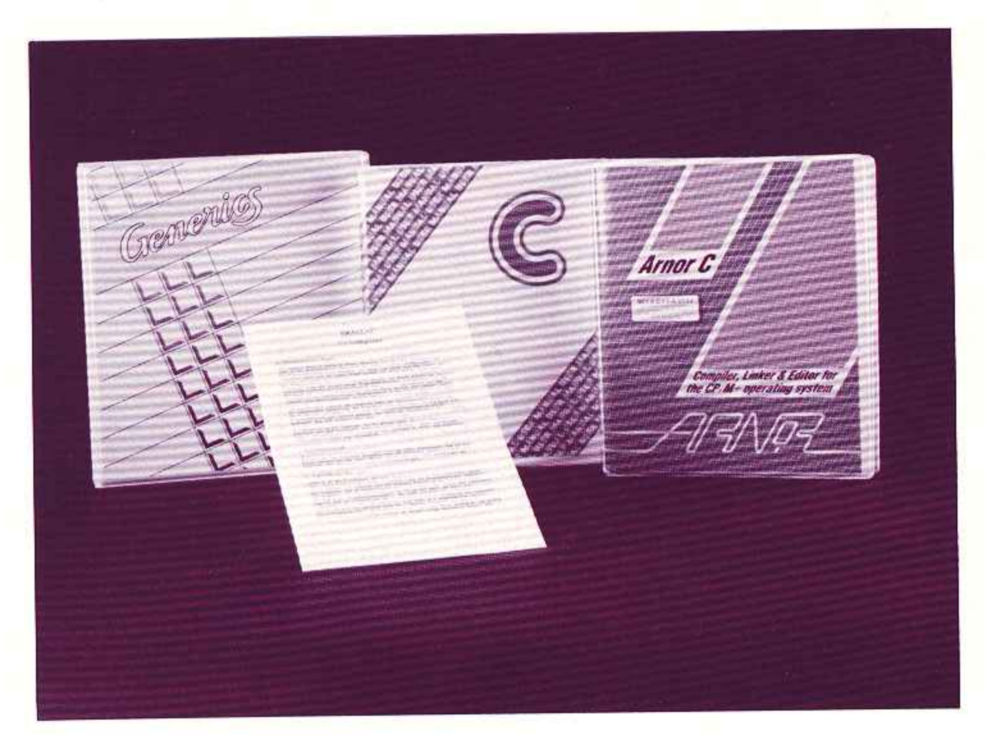
Procedure ShellSort (Var Class: StudentArray;
                     ClassSize: Integer);

Var Offset,          { Vergleichsoffset }
    Tausch,          { Wo war letzer Tausch ? }
    Limit,           { Anzahl zu vergleichender
                       Elemente }
    Line:  Integer;  { Aktuelle Line }

 Procedure Switch (Var Stu1, Stu2: Student);
 Var TempStu: Student;
 Begin                                          { Switch }
  TempStu:=Stu1;
  Stu1:=Stu2;
  Stu2:=TempStu;
 end;                                           { Switch }

Begin                                        { ShellSort }
 Offset:=ClassSize Div 2; { Vergleichsoffset auf
                            die Haelfte der Satz-
                            zahl setzen }
 While (Offset > 0) Do
 Begin { Schleife, bis Offset = 0 }
  Limit:=ClassSize-Offset;
  Repeat
   Tausch:=0; { Nicht tauschen bei diesem Offset }
   For Line:=1 To Limit Do
   If (Class[Line] > Class[Line+Offset]) Then
   Begin { austauschen }
    Switch (Class[Line],Class[Line+Offset]);
    Tausch:=Line;
   End;  { austauschen }
   { Sortierung nur bis dahin, wo letzer Tausch
     war }
   Limit:=Tausch-Offset;
  Until (Tausch=0);
  { Beim letzen Offset kein Tausch. Versuch mit
    halbiertem Offset }
  Offset:=Offset Div 2;
 End;
End;                                         { ShellSort }

{ Ende Shell.Inc                                              }
```



Viermal C

C-Compiler für CPC und Joyce

Haben Sie bis jetzt immer geglaubt, die geheimnissumwitterte Programmiersprache C gäbe es nur auf PCs und Großrechnern? Dann lassen Sie sich hier eines besseren belehren.

C ist eine Programmiersprache, die zusehends an Bedeutung gewinnt. Neben Turbo-Pascal ist es wohl die von Programmierern am meisten benutzte Sprache. Entwickelt wurde C Anfang der 70er Jahre von Dennis Ritchie und Brain Kernighan. Sie arbeiteten damals gerade an dem UNIX-Betriebssystem und suchten nach einer Möglichkeit, die Programmierarbeit zu vereinfachen. Die Systemprogramme von UNIX sind selbst größtenteils in C geschrieben, was die Leistungsfähigkeit der Sprache wohl am besten verdeutlicht. In Einheit mit UNIX kam C dann auch zu seiner Verbreitung.

Bald wurden auch für andere Systeme, wie beispielsweise CP/M, entsprechende C-Compiler entwickelt. Hierbei enstanden aber die verschiedensten Dialekte, die sich mehr oder weniger auf das Kernighan-und Ritchie- C, kurz K&R-C, bezogen. Das war natürlich kein Zustand. Hatte man einen Quelltext in C vorliegen, hieß das noch lange nicht, daß der eigene Compiler auch in der Lage war, diesen zu verarbeiten.

Anfang 1989 wurde man sich dann halbwegs einig, und das American National Standard Institute, kurz "ANSI", verabschiedete eine Sprachdefinition für C.

Da die C-Compiler, die wir Ihnen hier vorstellen möchten, schon um 1985 und früher entwickelt wurden, entsprechen diese dem ANSI-Standard nur in begrenztem Umfang. Da es sich hier aber meist um nur geringfügige Abweichungen handelt, sind die Anpassungen der Programme meist einfach zu realisieren.

Vorweg noch ein paar Bemerkungen: Wir haben die Compiler immer auf Grundlage der Erstellung einer ausführbaren Programmdatei(COM-Datei) verglichen. Hierfür wurden zwei Listings herangezogen: ein "leeres" Programm der Form:

```
/*Listing TEST.C*/
#include <stdio.h> (bzw. iolib.h)
main()
{
}
```

und das für C-Code halbwegs repräsentative Programm "COUNT.C", welches im Anschluß abgedruckt ist. Das abgedruckte Listing entspricht der Syntax des Small-C-Compilers von Kotulla. Um es mit den anderen Compilern nutzen zu können, muß es natürlich entsprechend angepaßt werden.

Beginnen wir mit dem Small-C-Entwicklungssystem. Dieses Paket stellt eine Einheit von Editor, C-Compiler, Assembler und Linker dar. Zusätzlich werden auf den insgesamt drei Disketten viele nützliche Utilities geliefert. Zu allen Programmen einschließlich dem Compiler, Assembler und so weiter wird der C-Quellcode mitgeliefert. Hierdurch hat man alle Möglichkeiten offen, selbst Modifikationen des Compilers sind unproblematisch. Anfänger erhalten umfangreiches Anschauungsmaterial in Form dieser Quelltexte.

Das Small-C-Entwicklungssystem

Nachteilig ist das Fehlen einer Fließkommabibliothek und die Übersetzung in den unflexibleren 8080-Code. Von Vorteil ist die Übersetzung in Assemblerquelltext und die dadurch mögliche Nachbearbeitung und Optimierung per Hand. Das Einfügen von Assemblerbefehlen in den C-Quelltext ist auch ohne weiteres zwischen den Anweisungen #asm und #endasm möglich.

Zum Erstellen eines Programmes sind insgesamt vier Schritte notwendig:

1. Erstellung des Programmtextes
2. Compilieren
3. Assemblieren
4. Linken

Für die Schritte zwei bis vier legt man sich am besten eine Submit-Datei an. Diese könnte beispielsweise folgendes Aussehen haben:

```
cc $1 -a -p
mac $1 -P
lnk $1 c.lib
```

Das zeitaufwendigste der drei Programme ist das Linkprogramm. Die Gesamtzeit für die Übersetzung liegt im noch als gut einzuschätzenden Bereich. Mit der Länge der ausführbaren Programme liegt das Small-C-Entwicklungssystem an dritter Stelle der hier verglichenen Compiler.

Der Compiler, wie auch die beiden anderen Programme, gibt zwar Klartextmeldungen aus, aber verrät leider nicht die Nummer der Zeile, in der der Fehler auftritt. Die Fehlersuche wird hierdurch etwas erschwert.

Folgendes ist nicht implementiert:

- Fließkommadatentypen
- mehrdimensionale Felder
- Strukturen und Unions
- Bit-Felder
- Zeigerarrays

Trotz dieser Nachteile steht aber dem

Programmierer noch eine genügend große Untermenge des Standard-C zur Programmentwicklung zur Verfügung. Das Small-C-Entwicklungssystem wird mit einem guten deutschen Handbuch geliefert.

Small-C(PD)

Die für Anfänger wohl attraktivste, da billigste Version eines C-Compilers ist dieses Small-C. Die amerikanische Originalversion ist in der Public-Domain, aber leider in der ursprünglichen Form auf dem CPC oder dem Joyce nicht lauffähig. Von Martin Kotulla wurde diese Version angepaßt und die Bildschirmmeldungen ins Deutsche übersetzt.
Beim Kauf erhält man eine Drei-Zoll-Diskette und ein deutsches Handbuch. Auf der Diskette befinden sich noch der notwendige Assembler ZMAC, der Linker ZLINK und ein Editor zur Bearbeitung des Quelltextes.
Der Sprachumfang des Compilers stellt eine Untermenge des Standard-C dar. Die Bibliotheksbezeichnungen sind leider nicht kompatibel gehalten. So nennt sich beispielsweise die notwendige Standard-Include-Datei hier nicht stdio.h sondern iolib.h. Aber das sind eigentlich nur Kleinigkeiten, die man sich schnell eingeprägt hat.
Es gibt auch eine Sonderfunktion für die Ein- beziehungsweise Ausgabe von binären Daten mit der Bezeichnung putb() beziehungsweise getb(). Besonders hervorheben sollte man die implementierte Fließkommaarithmetik. Auf folgende Dinge müssen Sie bei diesem Small-C verzichten:

- Statements:
  - for
  - do-while
  - switch-case
  - default
  - goto
  - mehrdimensionale Felder
  - Structuren und Unions
  - Zeiger auf Felder

Zur Programmcompilierung sollten Sie sich hier ebenfalls eine Submit-Datei anlegen, hier ein Beispiel wie diese für unser Beispiellisting aussehen müßte:

```
cc $1
zmac $1=$1
zlink $1=$1,iolib,printf1,args
```

Z80-Assemblerquelltext kann ebenfalls problemlos in den C-Quelltext eingebunden werden. Der Compiler übersetzt die C-Quelle in Z80-Assemblerquelltext. Dieser kann dann nötigenfalls noch per Hand optimiert werden. Anschließend übersetzt diesen der Assembler ZMAC in eine Objektdatei, die dann durch den Linker ZLINK zu einer COM-Datei gebunden wird. Die Größe der entstehenden COM-Datei richtet sich stark nach den zusätzlich eingebundenen Bibliotheken. Da viele Implementationen des ANSI-C fehlen, ist es wohl mit diesem Compiler am kompliziertesten, ein anderes Programm auf diesen umzusetzen. Zur Entwicklung eigener Programme und zum Erlernen der Sprache C ist dieser aber gut geeignet.

Hisoft C

Mit dem Hisoft C erhält man auf einer Drei-Zoll-Diskette ein Paket aus zwei C-Compilern nebst zugehörigem Editor. Einer davon ist als Besonderheit auf dem CPC von AMSDOS aus startbar. Selbst auf dem CPC 464 mit nur dem Kassettenlaufwerk ist dieser uneingeschränkt benutzbar. Als besonderes Bonbon für BASIC-Freaks ist eine kommplette Bibliothek der BASIC-Befehle vorhanden. Somit wird ein sehr schnelles Umlernen möglich. Erkauft wird das alles durch eine sehr geringe Kompatibilität zum ANSI-Standard.

Den Vergleich mit dem ANSI-C hält die CP/M-Variante schon eher stand. Abweichungen gibt es da beispielsweise bei der Nutzung der Argumentenübergabe aus der Kommandozeile. So muß, bevor die Argumente zur Verfügung stehen, als erstes eine Funktion namens cpm_cmd_line() aufgerufen werden. Leider hat auch dieses C keine Fließkommafunktionen. Die Syntax der vorhandenen Funktionen entspricht größtenteils der des ANSI-Cs. Zu beachten ist hier, daß Variablen nicht innerhalb des Funktionsnamens deklariert werden dürfen. Beispiel:
ANSI-C:

```
int funktion(char parameter)
{
.
.
}
```

Hisoft-C:

```
int funktion(parameter)
char parameter;
{
.
.
}
```

C-Sprachumfang nach dem ANSI-Standard

Dateibefehle:

*fopen	*freopen	fflush	fclose	remove	rename	*tmpfile
*tmpnam	setvbuf	setbuf	fseek	ftell	rewind	fgetpos

Ein- und Ausgabebefehle:

fprintf	printf	sprintf	vprintf	vfprintf	vsprintf	fscanf
scanf	sscanf	fgetc	*fgets	fputc	fputs	getc
getchar	*gets	putc	putchar	puts	ungetc	fread
fwrite						

Fehlerbehandlung::

clearerr	feof	perror				

Tests für Zeichenklassen und Umwandlung:

isalnum	isalpha	iscntrl	isdigit	isgraph	islower	isprint
ispunct	isspace	isupper	isxdigit	toupper	tolower	

Zeichenketten-Operationen:

*strcpy	*strncpy	*strcat	*strncat	strcmp	strncmp	*strchr
*strrchr	strspn	strcspn	*strpbrk	*strstr	strlen	*strerror

Mathematische Funktionen:

sin	cos	tan	asin	acos	atan	atan2
sinh	cosh	tanh	exp	log	log10	pow
sqrt	ceil	floor	fabs	ldexp	frexp	modf

Hilfsfunktionen:

atof	atoi	atol	strtod	strtol	strtoul	rand
srand	*calloc	*malloc	*realloc	free	abort	exit
atexit	system	*getenv	*bsearch	qsort	abs	labs
div	ldiv					

Globale Sprünge und Signale:

setjmp	longjmp	*signal	raise			

Zeit- und Datumsfunktionen:

clock	time	difftime	mktime	*asctime	*ctime	*gmtime
localtime	strftime					

Die Quelltextentwicklung können Sie mit dem mitgelieferten Texteditor vornehmen. Die Compilierung geschieht in einem Zuge mittels HC.COM. Findet der Compiler einen Fehler, so gibt er eine Klartextmeldung aus. Durch Drücken der Taste <E> kann nun auf Wunsch der Editor gestartet werden. Der Cursor rückt dabei nach dem automatischen Laden des Quelltextes selbständig auf die Position der Zeile, in der der Fehler auftrat.

Damit ist ein "Entwanzen" ähnlich wie unter TURBO-Pascal möglich. Das nach erfolgreicher Übersetzung entstandene Maschinenprogramm(.COM-Datei) ist an Kürze von den anderen vorgestellten Compilern nicht zu übertreffen.

Die ganze Sache wird aber von dem nur auf Englisch zu erhaltenen Handbuch getrübt. Solange man auf Fließkommaoperationen verzichtet, ist es mit diesem Compiler nach unseren Erfahrungen am einfachsten, einen Quelltext nach dem ANSI-Standard zum Laufen zu bekommen.

Arnor C

Ein sehr kompaktes Paket erhält man mit dem Arnor C. Der Editor bildet gleichzeitig eine Bedienoberfläche, von der aus alle anderen möglichen Aktionen gestartet werden können. Für den Editor besteht eine integrierte On-Line-Hilfe.

Den Übersetzungsvorgang startet man am einfachsten mit AC.COM. Dieses Utility übernimmt dann den Rest der Arbeit wie compilieren, linken und selbst den ersten Programmstart nach erfolgreicher Übersetzung.

Im Unterschied zu den anderen Compilern ist das Ergebnis der Programmentwicklung kein ausführbares Maschinenprogramm. Nur durch den Start mit Hilfe des Run-Time-Systems kann es zur Ausführung gebracht werden. Da also nicht an jeder Datei die Laufzeitbibliothek anhängt, wird natürlich Platz gespart. Möchte man aber ein ausführbares Programm, so kann man sich dies durch die Routine MAKECOM erstellen lassen. Die ausführbaren COM-Dateien fallen aber bei diesem Compiler außerordentlich groß aus.

Bei Arnor C wird es wohl am einfachsten sein, ein ANSI-C Programm umzusetzen, da auch eine Fließkommabibliothek implementiert ist. Beim CPC sollte man aber ein Zweitlaufwerk angeschlossen haben, da man nicht alle benötigten Dateien immer auf eine 178 kByte Diskettenseite bekommt.

Compiler	Small-C(Kotulla)	Small-C-Entwickl.	Highsoft-C	Arnor C
Größe von TEST.COM in kByte	4096	6272	4736	25856
Größe von COUNT.COM in kByte	9088	10112	7168	26624
Übersetzungsgeschwindigkeit	sehr gut	gut	gut	mäßig
Bezugsmöglichkeit	Fa. Weeske	Fa. Wiedmann	Fa. Wiedmann	Fa. Wiedmann Fa. Weeske
Preis in DM	20.00	99.00	199.50	199.00/225.00

Nun noch einige Hinweise zu unserem Beispiellisting COUNT.C. COUNT zählt das Auftreten der Codes 0 bis 255 in einer beliebigen Datei und gibt diese je nach Wunsch der Größe nach geordnet oder geordnet nach der Häufigkeit des Auftretens aus.

Das Programm besteht aus sechs teilen.

1. Hauptprogramm: Erledigt die Argumentenübergabe, das Öffnen und Schließen der Datei und die Auswertung der Schalteroption aus der Kommandozeile. Außerdem wird der Aufruf der Unterroutinen koordiniert. Im Fehlerfall wird *badarg() aufgerufen.*

2. badarg(): Wird im Fehlerfall vom Hauptprogramm aufgerufen. Es gibt einen Syntaxhinweis aus, und beendet das Programm vorzeitig mit *exit().*

3. count(): ist eine Prozedur zur Registrierung der Häufigkeit eines Zeichens. Die Bytes(Zeichen) werden einzelnen aus der Datei gelesen und ihr Wert dient als Index für ein Feld aus 256 Bytes. Das jeweils ausgewählte Feldelement wird um eins erhöht.

4. ausgab(): gibt die Anzahl der jeweiligen Zeichen in Reihenfolge ihrer Größe aus.

5. prchar(): Falls ein Zeichen im Bereich von 32 bis 127 liegt, also ein darstellbares Zeichen ist, wird es auf den Bildschirm ausgegeben. Diese Prozedur wird von *ausgab() und ausgroes() aufgerufen.*

6. ausgroes(): Diese Routine gibt die Zeichen, nach der Häufigkeit ihres Auftretens geordnet, auf dem Bildschirm aus. Hierzu werden zwei Hilfsfelder angelegt. Ein Feld wird zur Registrierung der Reihenfolge der Zeichen nach der Häufigkeit, und das andere Feld zur Registrierung der Anzahl der jeweiligen Codes benötigt.

Im Quelltext wird vor allem die vom ANSI-Standard stark abweichende Argumentenübergabe aus der Komandozeile, und das Lesen einer binären Datei verdeutlicht.

Nach dem Start des Programms aktiviert als erstes die Funktion setargs() die Argumentenübergabe. Anschließend erfolgt die Übergabe der Argumente mit der Funktion getarg(). Im Fehlerfall gibt diese den Wert -1 (EOF) zurück, worauf hin dann eine Funktion badarg() aufgerufen wird. Diese gibt einige Syntaxhinweise für die Komandozeile aus, und beendet das Programm vorzeitig mit exit().

Wurden die Argumente fehlerfrei übergeben, wird nun die gewünschte Datei mit fopen() zum Lesen geöffnet(Argument r). Die nun folgende Funktion count() zählt das Auftreten der verschiedenen Codes von 0-255 und legt die Werte in einem Feld ab. Nach deren Beendigung wird nach Auswertung der Auswahloption die jeweilige Ausgabefunktion gewählt. Am schnellsten arbeitet hierbei ausgab(). Hier werden in einer Zählschleife die einzelnen Werte ausgegeben.

Die Funktion ausgroes() führt zuerst einen etwas zeitaufwendigen Sortieralgorithmus aus, der die Werte der Größe nach in einem Feld ablegt. Dieser ist recht einfach, mittels einer Schleife wird jeweils der größte Wert aus dem Feld herausgesucht und dann mit steigendem Index auf einem zweiten Feld abgelegt. Daraufhin wird der Wert auf Null gesetzt. Am Ende dieses Algorithmus sind alle Komponenten des Ursprungsfeldes gleich Null.

Anschließend wird wiederum mittels einer Zählschleife das Feld auf dem Bildschirm ausgegeben. In diesem Programm wurden absichtlich keine komplizierten Algorithmen eingesetzt, da es auch für Anfänger leicht durchschaubar sein sollte.

jg

Literatur: [1] TOOLBOX 12'90, Ur-C im Abseits, DMV-Verlag

```
/***********************************************
*                                              *
*            Listing COUNT.C                   *
*     (c) by j.g. & CPC Intenational           *
*    Programm zur Ermittlung der Haeufigkeit   *
*    von Zeichen in einer beliebigen Datei     *
*                                              *
***********************************************/
#include iolib.h     /*Bibliothek mit allen
                       noetigen Grundfunktionen*/
#include printf1.h   /*Bibliothek zur formatierten
                       Ausgabe von Strings und
                       Integerwerten*/
#include args.h      /*Bibliothek mit Funktionen
                      zur Uebergabe von Argumenten
                      aus der Komandozeile*/
#define EOF -1       /*Definition der Konstante
                       fuer Dateiende*/

main()
{
        int *fp,      /*Filepointer*/
            byte,     /*gelesenes Zeichen*/
            tab[256];/*Feld zum Zaehlen fuer 256
                       moegliche Zeichen*/
        char name[16], /*Stringvariable fuer
                         Dateiname*/
             o[2];     /*Auswahloption*/
        setargs();     /*Initialisierung der
                         Argumentenuebergabe*/

/*Uebergabe des 1.Arguments mit Fehlerauswertung,
wenn der Rueckgabewert EOF ist wurde kein
Argument uebergeben und das Programm wird nach
der Fehlermeldung mit exit vorzeitig verlassen*/

        if(getarg(1,o,2)==EOF) badarg();
        if(getarg(2,name,16)==EOF) badarg();

        fp=fopen(name,"r"); /*Datei oeffnen*/

/*Aufruf von count, als Argumente werden in diesem
Fall Zeiger uebergeben, fp zeigt auf den
Dateideskriptor, tab zeigt auf den
Anfang des Feldes tab[]*/

                count(fp,tab);

/*Aufruf der gewuenschten Ausgaberoutine, als
Argument wird ein Zeiger auf den Beginn des
Feldes tab[] uebergeben, trifft keine Option
zu wird ebenfalls badarg aufgerufen*/

        if(upper(o[0])=='R')
                ausgab(tab);
        else if(upper(o[0])=='H')
                ausgroes(tab);
        else badarg();

        fclose(fp); /*Schliessen der Datei*/

}
/*Ende des Hauptprogramms*/

/*Ausgabe eines Syntaxhinweises bei
falschen Argumenten*/
badarg()
{
        printf("Fehler in der Eingabezeile\n");
        printf("Aufruf:\n");
        printf("Ausgabe in Reihenfolge   : ");
        printf(" count r Datei\n");
        printf("Ausgabe nach Haeufigkeit: ");
        printf(" count h Datei\n");
        exit(0);    /*Programm vorzeitig beenden*/
}

/*Prozedur zur Registrierung der Haeufigkeit
eines Zeichens*/

count(dd,feld)
int *dd,   /*Enthaelt den Zeiger auf den
                Dateideskriptor*/
    *feld; /*Enthaelt den Zeiger auf
                das Variablenfeld*/
{
        int byte,c;  /*c-Schleifenzaehler*/
        c=0;

        /*alle Werte des Feldes auf Null setzen*/
        while(c!=256)        /*Abbruch wenn c=255*/
                feld[c++]=0;
                /*Feld mit Index c auf Null setzen,
                dann c incrementieren*/


        while((byte=getb(dd))!=EOF)  /*Lesen und
                                Test auf Erreichen
                                des Dateiendesi*/
                feld[byte]++;  /*Feldvariable mit
                        Index byte incrementieren*/

}

/*formatierte Bildschirmausgabe in
Reihenfolge der Codes*/

ausgab(feld)
int *feld;
{
        int c,    /*Schleifenzaehler*/
            z;    /*Zeilenzaehler*/

        c=z=0;
        while(c!=256){      /*Abbruch bei c=256*/
           if(z++ ==21){    /*Nach Ausgabe  */
                getchar(); /*von 21 Zeilen */
                z=0;        /*Warten auf    */
                }           /*Tastendruck   */
           printf("Dez: %3d Hex: %2x Z: ",c,c);
           prchar(c);
           printf("  Anzahl: %d\n",feld[c++]);
           }
}

/*Ausgabe eines Codes als Zeichen, falls es
darstellbar ist*/

prchar(z)
int z;
{
        if(z>31 & z<128)
                putchar(z);
}
/*formatierte Bildschirmausgabe geordnet
nach der Haeufigkeit des Auftretens*/

ausgroes(feld1)
int *feld1;
{
        int c,cc,  /*Schleifenzaehler*/
            z,     /*Zeilenzaehler*/
            max,   /*momentan groesster Wert*/
            ind,   /*Index des groessten Wertes*/
            feld2[256], /*Tabelle mit
                        nach Haeufigkeit
                        sortierten Werten*/
        feld3[256]; /*Tabelle fuer Haeufigkeiten*/
        c=cc=0;
        while(c!=256){
                max=feld1[cc]; /*1.Wert merken*/
                ind=0;
                while(++cc!=256)

                        /*1.Wert mit 2.
                        vergleichen, wenn groesser
                        merken*/
                        if(max<feld1[cc]){
                                max=feld1[cc];
                                ind=cc;
                                };
                /*groessten Wert im Feld mit Index
                c ablegen*/
                feld2[c]=ind;
        feld3[c++]=max; /*Anzahl merken*/
                feld1[ind]=cc=0;
                }

        /*Bildschirmausgabe*/
        c=z=0;
        while(c!=256){      /*Abbruch bei c=256*/
           if(z++ ==21){    /*Nach Ausgabe  */
                getchar(); /*von 21 Zeilen */
                z=0;        /*Warten auf    */
                }           /*Tastendruck   */
           printf("Dez: %3d Hex: ",feld2[c]);
           printf("%2x Z: ",feld2[c]);
           prchar(feld2[c]);
        printf(" Anzahl: %d\n",feld3[c++]);
           }
}
```

Probleme mit INLINE?

Anzeige der Registerinhalte in TURBO-Pascal

Benutzen Sie in Ihrem TURBO-Pascal-Programm die INLINE-Anweisung, und haben Sie Schwierigkeiten beim Austesten Ihres Programms? Hier hilft Ihnen die Prozedur Z80-INFO.

Die meisten Programme lassen sich mit Hilfe einer höheren Programmiersprache schneller entwickeln als in Assembler. Programmerstellung in Assembler ist da immer etwas zeitraubender. Aber es gibt auch Beispiele, wo eine von höheren Programmiersprachen nicht zu erreichende Ausführungsgeschwindigkeit gefragt ist. Das ist dann die Stunde des "Inline-Codes": Man bettet Assembler-Programmroutinen in ein Hochsprachenprogramm ein. Was in BASIC der POKE-Befehl in Zusammenhang mit dem CALL ist, löst man in TURBO-Pascal mit der INLINE-Anweisung.

Hiermit wird dem Programmierer die höchstmögliche Ausführungsgeschwindigkeit geboten. Man denke nur an Grafikroutinen, bei denen immer Höchstgeschwindigkeit gefragt ist. Ein Beispiel für die Übersetzung eines Wertevergleichs, um ein Punktsetzen außerhalb des Bildschirms zu vermeiden, wird in der Abbildung gezeigt. Aus dem relativ kurzen Quelltext werden 88 Bytes Maschinenprogramm. Als besonders zeitraubend sind auch die vier Aufrufe der Laufzeitbibliothek anzusehen. In Assembler programmiert, kommt man bei diesem Beispiel mit etwa 17 Bytes aus.

Dieses sehr viel kürzere Programm kann dann natürlich auch entsprechend schneller abgearbeitet werden. Wer schon einmal mit INLINE-Anweisungen gearbeitet hat, weiß aber auch, wie schwierig und zeitaufwendig es ist, semantische Fehler im Maschinencode zu finden (Semantische Fehler sind Fehler im gedanklichen Aufbau eines Programmes – syntaktische sind dagegen Fehler welche der Grammatik der Sprache widersprechen, die letzteren findet der Compiler bei der Übersetzung). Eine Fehlersuche ohne Hilfsmittel ist nahezu unmöglich.

Während man die semantischen Fehler im Pascal-Code durch Einfügen von Schreibanweisungen finden kann, geht dies bei den INLINE-Anweisungen nicht. Hier lassen sich keine zusätzlichen Befehle aus dem Befehlsvorrat von Pascal einfügen.

Fein raus sind lediglich diejenigen, die vorher ihre Inlines mit Hilfe eines Assemblers austesten können. Doch auch sie werden Schwierigkeiten bei der Übertragung in den Pascalcode haben, es sei denn, sie besitzen einen INLINE-Generator. Jetzt kann Ihnen jedoch geholfen werden.

Während des Compilierens binden Sie einfach mittels Include-Anweisung das Modul Z80-INFO.INC ein. An die zu testenden Stellen plaziert man nun die INLINE-Anweisung

`$cd/Z80_INFO/(*Call Z80_INFO*).`

Nun erhält man auf dem Bildschirm einen vollständigen Überblick über den Zustand der Z80-Register. Alle werden angezeigt, die Hauptregister bitweise und hexadezimal, Indexregister, Programmzähler und Stackpointer nur hexadezimal (Der Programmzähler zeigt natürlich die Stelle, an der zur Procedur Z80_Info verzweigt wurde). Anschließend wird gefragt, ob das Programm abgebrochen oder fortgesetzt werden soll. Dies ist notwendig, um beispielsweise aus Endlosschleifen herauszukommen, und macht einen RESET fast überflüssig.

Entscheidet man sich für die Fortführung des Programmes, so wird der Prozessor wieder in seinen alten Zustand versetzt, den er unter anderem durch die Ausgabe der Registerinhalte verloren hatte. Eines muß man jedoch dabei beachten: Das Einfügen eines Aufrufes der Prozedur Z80_INFO beansprucht drei Bytes (CALL-Befehl plus Adresse). Dies muß bei allen Sprung oder besser Verzweigungsbefehlen berücksichtigt werden. Eine Empfehlung: Man schafft an allen möglicherweise zur Registerausgabe interessanten Stellen Platz, indem man drei NOPs (No

```
Aus dem Pascal-Quelltext:

begin
if(x >= 0) and (x<640) and (y>=0)
and (y< 200) then punkt(x,y);
end;

wird folgender Maschinencode
(hier disassembliert)       :

      POP IY
      POP HL
      LD  (X),HL
      POP HL
      LD  (Y),HL
      PUSH     IY
      LD  HL,(Y)
      PUSH     HL
      LD  HL,0
      POP DE
      CALL     06A5H                 ;1. Aufruf
      PUSH     HL
      LD  HL,(Y)
      PUSH     HL
      LD  HL,0280H
      POP DE
      CALL     06E0H                 ;2. Aufruf
      POP DE
      LD  A,L
      AND E
```

```
LD  L,A
PUSH     HL
LD  HL,(X)
PUSH     HL
LD  HL,0
POP DE
CALL     06A5H                 ;3. Aufruf
POP DE
LD  A,L
AND E
LD  L,A
PUSH     HL
LD  HL,(X)
PUSH     HL
LD  HL,0C8H
POP DE
CALL     06E0H                 ;4. Aufruf
POP DE
LD  A,L
AND E
LD  L,A
BIT 0,L
JP  Z,2163H                    ;zum RET-Befehl
LD  HL,(Y)
PUSH     HL
LD  HL,(X)
PUSH     HL
CALL     PUNKT
RET
```

operation, Code &00) einfügt. Diese können dann bei Bedarf durch den Aufruf von Z80_INFO ersetzt werden. Läuft das Programm dann zufriedenstellend, kann man in einem zweiten Anlauf die NOPs wieder entfernen. Dabei sollte man darauf achten, daß man die Verzweigungsbefehle wieder um drei Bytes herauf beziehungsweise heruntersetzt. Hier noch einige Anmerkungen zum Programm:

1. Benötigt man nicht die komplette Ausgabe, so kann durch Einfügen von der Kommentarzeichen "{" und "}" in der Prozedur *ausgabe* diese einfach variiert werden. Man kann so zum Beispiel auf die binaäre Ausgabe verzichten oder sich den Zustand der Hintergrundregister nicht anzeigen lassen. Dies schafft Übersicht, und man gewinnt Zeit.

2. Nicht nur für das Austesten von Inline-Anweisungen ist das Programm zu gebrauchen, sondern auch zum Lernen und Verstehen der einzelnen Z80-Befehle. Weiß man nicht genau, was ein Befehl bewirkt, so kann man sich die Wirkung mit Hilfe von Z80_INFO leicht veranschaulichen.

Ein Anwendungsbeispiel für die Prozedur Z80_INFO zeigt das Programm Z80_DEMO.PAS. Viel Spaß bei der Arbeit "hart am Bit".

H. Hagemeyer,jg

Z 80 - Status

Erstregister:

A	SZ-H-PNC	B	C	D	E	H	L
1111.1111	1100.1010	1111.0000	0000.0000	0010.0000	1101.1110	1000.1101	1100.1001
$FF	$CA	$F0	$00	$20	$DE	$8D	$C9

Zweitregister:

A	SZ-H-PNC	B	C	D	E	H	L
1011.1011	1010.1100	0000.0000	0000.0000	0111.1011	0110.0100	0101.1110	1000.1101
$BB	$AC	$00	$00	$7B	$64	$5E	$8D

Index- und Adressregister

IX	IY	SP	PC
$79DA	$95DF	$F580	$95A1

Weiter / Halt (W/H) ?

So zeigt Z80_INFO die aktuellen Registerinhalte an.

```
(*-----------------------------------------------*)
(*                  Z80DEMO.PAS                  *)
(* Demonstriert die richtige Anwendung der       *)
(* Prozedur Z80_INFO in INLINE - Anweisungen.    *)
(* Dient gleichzeitig als Testprogramm.          *)
(* (c)1991 H.Hagemeyer & CPC International       *)
(*-----------------------------------------------*)
PROGRAM inline_demonstration;

(*$I Z80-INFO.INC Einbinden der Prozedur Z80_INFO*)


CONST WERT = $FF;

BEGIN
  INLINE(
       $3E/WERT/      (* LD A,WERT ;Demo Schleife *)
(*L1*) $CD/Z80_INFO/ (* CALL Z80_INFO ;Ausgabe des
                         Prozessor - *)
                      (* status (A=FF bis 01h)*)
       $3D/       (* DEC A ;A <-- A - 1 (A = FEh) *)
       $E6/$07/   (* AND 07h ;Ausblenden der
                     4 oberen*)
                           (* ;Bit's (A = 0Eh *)
       $C2/*-7/     (* JP NZ,L1 ;Schleifenanfang *)
       $CD/Z80_INFO/ (* CALL Z80_INFO ;ZERO - FLAG
                         = 1 *)
       $3E/WERT/      (* LD A,WERT *)
       $47/           (* LD B,A *)
       $CD/Z80_INFO/  (* CALL Z80_INFO ;B = FFh *)
       $48/           (* LD C,B *)
       $CD/Z80_INFO/  (* CALL Z80_INFO ;C = FFh *)
       $57/           (* LD D,A *)
       $CD/Z80_INFO/  (* CALL Z80_INFO ;D = FFh *)
       $5F/           (* LD E,A *)
       $CD/Z80_INFO/  (* CALL Z80_INFO ;E = FFh *)
       $67/           (* LD H,A *)
       $CD/Z80_INFO/  (* CALL Z80_INFO ;H = FFh *)
       $6F/           (* LD L,A *)
       $CD/Z80_INFO/  (* CALL Z80_INFO ;L = FFh *)
       $DD/$21/WERT/WERT/ (* LD IX,WERT*)
       $CD/Z80_INFO/  (* CALL Z80_INFO ;IX = FFFFh
       *)
       $FD/$21/WERT/WERT/ (* LD IY,WERT,WERT *)
       $CD/Z80_INFO/  (* CALL Z80_INFO ;IY = FFFh *)
       $E5/           (* PUSH HL *)
       $CD/Z80_INFO/  (* CALL Z80_INFO ;Stapelzeiger
       - 1 *)
       $E1/           (* POP HL ;Stapelzeiger + 1 !! *)
       $08/           (* EX AF,AF' ;Austausch der *)
       $D9/           (* EXX ;Register *)
       $CD/Z80_INFO (* CALL Z80_INFO;Register-
                         austausch *)
   );
END.
```

```
(*-----------------------------------------------*)
(*              Z80-INFO.INC                     *)
(*ein Modul zum Austesten von Inline-Anweisungen *)
(*        der Sprache TURBO PASCAL               *)
(*   (c)1991 H. Hagemeyer & CPC International    *)
(*-----------------------------------------------*)
PROCEDURE Z80_INFO;

TYPE str1 = STRING [10];

(*---------------------------------------------
Die Hauptregister des Z80 - Prozessors A, B, C, D,
H, L und das Flagregister F sind als Byte
definiert, ebenso die Hintergrundregister A', B',
...
Die Reihenfolge der Definitionen darf nicht
veraendert werden,da in den Inline - Anweisungen
ausgenutzt wird, das F hinter A, C hinter B etc.
liegt !
Die Indexregister IX und IY sind als Integer
definiert, welche 2 Bytes belegen, ebenso der Stack
- Pointer (Stapel - Zeiger SP) und der Programm -
Counter (Programm - Zaehler PC).
-----------------------------------------------*)
VAR
    A, F, B, C, D, E, H, L, (* Hauptregister *)
    As,Fs,Bs,Cs,Ds,Es,Hs,Ls : BYTE;
    (*Hintergrundregister des Z 80 Prozessors*)
    IX,IY,SP,PC : INTEGER;
    (*Index- und Adressregister *)
(*-----------------------------------------------*)

(*
Gibt den Hex.-Wert, eines Byte als String zurueck
*)

FUNCTION hexbyte (B : BYTE) : str1;
                         (* Gibt den Hexcode eines
                         Nibbles (4 Bit) zurueck) *)
```

```
FUNCTION hexteil (x : BYTE): CHAR;
  CONST null = 48;         (* ASCII von 0 *)
        A = 55;            (* " von A *)
  BEGIN
     CASE x OF
        0.. 9 : hexteil := Chr(x + null); (* Ziffer
        0 .. 9 *)
       10..15 : hexteil := Chr(x + A ); (* Ziffer A
       .. F *)
     END;
  END (* hexteil *);

BEGIN (* hexbyte *)
  hexbyte := hexteil(B SHR 4) + hexteil(B AND $0F)
END;
(*----------------------------------------------*)

(* Gibt den Hex.-Wert zweier Bytes als String
zurueck *)

FUNCTION hexint (i : INTEGER) : str1;
BEGIN
hexint := hexbyte(i SHR 8) + hexbyte(i AND $00FF);
END;
(*----------------------------------------------*)
(* Gibt den Binaerwert eines Bytes als String
zurueck *)

FUNCTION binaerbyte (A : BYTE) : str1;
VAR i : BYTE; s : str1;
BEGIN
  s := ''; i := $80;
  REPEAT
    IF ((i AND A) <> 0) THEN s := s + '1' (*
    Ausblenden mit i *)
    ELSE s := s + '0';
    IF i = $10 THEN s := s + '.'; (* In der Mitte
    einen . *)
    i := i SHR 1           (* naechstes Bit *)
  UNTIL i = $00;           (* Herausgeschoben *)
  binaerbyte := s + ' ';   (* Blank fuer Ausgabe
  anhaengen *)
END;
(*----------------------------------------------*)
(* Ausgabe auf dem Bildschirm der einzelnen
Register. A, F, B, C, ... sind *)
(*als lokale Variablen (inerhalb der Procedure
Z80_info) definiert.*)

PROCEDURE ausgabe;
VAR ch : CHAR;
BEGIN
  ClrScr; Write('':32); WriteLn('Z 80 - STATUS');
  WriteLn;
  WriteLn('Erstregister:'); WriteLn;
  WriteLn('    A        SZ-H.-PNC       B          C',
          '                 D              E ',
          '               H              L');
  WriteLn;
  Write(binaerbyte(A),binaerbyte(F));
  Write(binaerbyte(B),binaerbyte(C));
  Write(binaerbyte(D),binaerbyte(E));
  Writeln(binaerbyte(H),binaerbyte(L));


  WriteLn('$' + hexbyte(A): 6,'$' +
  hexbyte(F):10,'$' + hexbyte(B):10,
          '$' + hexbyte(C):10,'$' +
          hexbyte(D):10,'$' + hexbyte(E):10,
          '$' + hexbyte(H):10,'$' + hexbyte(L):10
  WriteLn; WriteLn; WriteLn('Zweitregister');
  WriteLn;
  WriteLn('    A        SZ-H.-PNC       B          C',
          '                 D              E ',
          '               H              L');
  WriteLn;
  Write(binaerbyte(As),binaerbyte(Fs));
  Write(binaerbyte(Bs),binaerbyte(Cs));
  Write(binaerbyte(Ds),binaerbyte(Es));
  Writeln(binaerbyte(Hs),binaerbyte(Ls));



  WriteLn('$' + hexbyte(As): 6,'$' +
  hexbyte(Fs):10,'$' + hexbyte(Bs):10,
          '$' + hexbyte(Cs):10,'$' +
          hexbyte(Ds):10,'$' + hexbyte(Es):10,
          '$' + hexbyte(Hs):10,'$' +
            hexbyte(Ls):10);
  WriteLn; WriteLn;
  WriteLn('Index- und Adressregister');
  WriteLn;
  WriteLn('     IX          IY          SP          PC');
  WriteLn;
  WriteLn('$' + hexint(IX):7,'$' +
  hexint(IY):10,'$' + hexint(SP):10,
          '$' + hexint(PC):10);
  WriteLn; WriteLn;
  Write('Weiter / Halt (W/H) ? ');
  REPEAT Read(Kbd,ch);
  ch := UpCase(ch);
  UNTIL ch IN ['H','W'];
  IF ch = 'H' THEN Halt; (* Programm abbrechen *)
END;
(*----------------------------------------------*)
BEGIN (* Z80_info *)
  INLINE(
                   (* Registerinhalte in die
                   entsprechenen Variablen bringen: *)
    $ED/$53/E/   (* LD (E),DE ;Lade D und E mit Reg.
                   DE *)
    $22/L/       (* LD (L),HL ;Lade H und L mit
                   Reg. HL *)
    $ED/$43/C/   (* LD (C),BC ;Lade B und C mit
                   Reg. BC *)
    $F5/         (* PUSH AF    ;Flags u. A *)
    $E1/         (* POP HL     ;nach HL bringen *)
    $22/F/       (* LD (F),HL ;Lade (A) u. (F) mit
                   HL = AF ! *)
    $08/         (* EX AF,AF' ;Tausche Register
                   AF mit AF' *)
    $D9/         (* EXX ; " " *)
    $ED/$53/Es/(* LD (Es),DE ;Lade Ds und Es mit
                   Reg. DE' *)
    $22/Ls/      (* LD (Ls),HL ;Lade Hs und Ls mit
                   Reg. HL' *)
    $ED/$43/Cs/(* LD (Cs),BC ;Lade Bs und Cs mit
                   Reg. BC' *)
    $F5/         (* PUSH AF ;Flags und Reg. A nach *)
    $E1/         (* POP HL ;HL bringen *)
    $22/Fs/      (* LD (Fs),A ;Lade (As) u.(Fs) mit
                   Reg. HL *)
    $DD/$22/IX/ (* LD (IX),IX ;Lade (IX) mit Reg.
                   IX *)
    $FD/$22/IY/ (* LD (IY),IY ;Lade (IY) mit Reg.
                   IY *)
    $ED/$73/SP/ (* LD (SP),SP ;Lade (SP) mit Reg.
                   SP *)
    $E1/         (* POP HL ;Rueckkehradresse holen *)
    $E5/         (* PUSH HL ;und retten *)
    $22/PC/ (* LD (PC),HL ;Lade (PC) mit Reg. HL *)
    $CD/ausgabe/ (* CALL AUSGABE *)
   (* Da bei der Ausgabe die Register veraendert
   werden, wird im folgenden der urspruengliche
   Zustand wiederhergestellt. *)
    $2A/Fs/ (* LD HL,(Fs) ;Lade HL mit Speicher As
                   u. Fs *)
    $E5/         (* PUSH HL ; und *)
    $F1/         (* POP AF ;bringe dieses nach AF *)
    $ED/$5B/Es/ (* LD DE,(Es) ;Lade DE mit Speicher
    Ds u. Es *)
    $2A/Ls/      (* LD HL,(Ls) ;Lade HL mit
                   Speicher Hs u. Ls *)
    $ED/$4B/Cs/ (* LD BC,(Cs) ;Lade BC mit Speicher
                   Bs u. Cs *)
    $08/         (* EX AF,AF' ;Tausche Register AF
                   mit AF'*)
    $D9/         (* EXX ; " " *)
    $2A/F/       (* LD HL,(F) ;Lade HL mit
                   Speicher A u. F *)
    $E5/         (* PUSH HL ; und *)
    $F1/         (* POP AF ;bringe dieses nach AF *)
    $ED/$5B/E/ (* LD DE,(E) ;Lade DE mit
                   Speicher D u. E *)
    $2A/L/ (* LD HL,(L) ;Lade HL mit Speicher
                   H u.L *)
    $ED/$4B/C/ (* LD BC,(C) ;Lade BC mit Speicher
                   B u. C *)
    $DD/$2A/IX/ (* LD IX,(IX) ;Lade IX mit Speicher
                   IX *)
    $FD/$2A/IY/ (* LD IY,(IY) ;Lade IY mit Speicher
                   IY *)
    $ED/$7B/SP (* LD SP,(SP) ;Lade SP mit Speicher
                   SP *)
  );
END;
```



AMS-Line

Insider-Informationen

In dieser Rubrik stellen wir für Sie die Direktverbindung mit den Fachleuten her. Heute erfahren Sie von der Firma Weeske etwas über BTX auf dem CPC und was dieser Dienst der Telekom Ihnen bringen kann.

Die BTX-Angebotspalette ist schon heute außergewöhnlich groß und umfaßt über eine halbe Million Bildschirmseiten. Von aktuellen Nachrichten aus Politik, Wirtschaft und Sport, über den Wetterbericht des deutschen Wetterdienstes, Lotto- und Totozahlen, Urlaubsangebote von Reiseveranstaltern, Fahr- und Flugplanauskünfte, lokale Veranstaltungskalender mit Vereins-, Theater- und Konzertterminen, Verkaufsangebote von Kaufhäusern und Versandhandelsunternehmen, Immobilienangebote von Maklern, Stellenangebote von Arbeitsämtern und Firmen, bis hin zum elektronischen Telefonbuch.

Doch dies ist nur eine Stärke des BTX-Dienstes, denn BTX ist ein Dialog-System. In der praktischen Anwendung heißt das: Der BTX-Benutzer kann mit jedem anderen BTX-Teilnehmer kommunizieren. Jeder BTX-Teilnehmer kann mit angeschlossenen Rechnern der Anbieter Kontakt aufnehmen. Hier stehen Buchungs- und Bestellsysteme, Informationssysteme und vieles mehr zur Verfügung.

Hier einige Beispiele:

– *Bestellung per BTX:*

Vom Schreibtisch oder Wohnzimmersessel aus und mit sofortiger Auftragsbestätigung kann beispielsweise beim Versandhaus geordert werden, können Eintrittskarten bei der Theaterkasse oder Bücher und Schallplatten bei Buchclubs und Verlagen bestellt werden.

– *Überweisung per BTX:*

Statt sich in Banken und Sparkassen in die oft langen Warteschlangen vor den Schaltern einzureihen, kann der BTX-Teilnehmer seine Bankgeschäfte zu Hause abwickeln: Überweisungen veranlassen, Daueraufträge einrichten, Kontoauszüge einsehen. Dies gilt auch für das Postgirokonto.

– *Kontakte über BTX:*

Über "Dialogsysteme" läßt sich anonym "smalltalken", fachsimpeln und auch flirten. So lernt man neue Leute kennen – allerdings entstehen je nach Anbieter des Dialogprogramms unterschiedliche Kosten.

– *Spielen mit BTX:*

Auch dies gibt es im BTX-Angebot: Kurzweil und Spaß per BTX. Gruppen-Abenteuer und Glücksspiele stehen ebenso auf dem Programm wie knifflige Denkaufgaben.

Dies sind nur einige Beispiele aus einer Vielzahl von Möglichkeiten, die das BTX-System dem privaten Teilnehmer bietet – zu jeder Tages-und Nachtzeit. Denn BTX kennt keinen Feierabend.

BTX – wie es funktioniert:

Wollen Sie BTX mit Ihrem Computer benutzen so benötigen Sie als CPC-Nutzer das früher von Siemens produzierte Schneider-BTX-Modul. Dieses wird noch von einigen Händlern aus Restbeständen vertrieben und ist im Vergleich zu Hardware-Decodern für den PC nahezu unverschämt billig, obwohl es die gleiche Leistung bringt. Für Benutzer eines Joyce gibt es den Wiedmann-BTX-Dekoder. Der Zugang zum BTX-System wird über die Telefonleitung hergestellt. Hierzu installiert die Post eine BTX-Anschlußbox "DBT-03", die BTX-Gerät und Telefon miteinander verbindet, automatisch das System anwählt und die gewünschte Information über die Leitung transportiert.

BTX – wie man damit umgeht:

Mit BTX umzugehen ist schnell gelernt. Und so wird's gemacht: BTX-Anwahltaste drücken, das persönliche Kennwort eingeben, und schon ist der Teilnehmer mit dem BTX-Dienst verbunden. Über systematisch aufgebaute Inhalts-, Schlagwörter- und Anbieterverzeichnisse (Suchbaummethode) findet der BTX-Benutzer Schritt für Schritt die Seite im BTX-Angebot, die ihn interessiert.

BTX – was es kostet:

BTX ist ein preiswertes Medium: Die einmalige Anschlußgebühr beträgt 65,– DM, die monatliche Grundgebühr 8,– DM. Der Rest hängt von der Nutzung ab. Die Verbindungsgebühren werden nach dem Telefon-Nahtarif berechnet. Dazu können noch geringe Entgelte kommen, die einige Anbieter für spezielle Informationen und Dienste verlangen.

BTX – wie sicher es ist:

Umfangreiche Schutzmaßnahmen garantieren die Sicherheit Ihrer persönlichen Daten. Schon bei der Anwahl des BTX-Dienstes werden individuelle Kennungen aus der Anschlußbox und Ihr persönliches Kennwort geprüft. Die Verwaltung und Sicherung Ihres persönlichen Kennwortes bestimmen Sie selbst. So wird sichergestellt, daß kein Dritter Ihren BTX-Anschluß mißbräuchlich benutzt. BTX ist also so sicher wie Ihr Telefon.

(Damasckke/jg)



100 DM für 1 kByte

Auf ein Neues!

Ausgeglichenheit ist diesen Monat angesagt. Unter den insgesamt sechs neuen Programmen stellen wir Ihnen genau drei Anwendungen und drei Spielprogramme vor. Achtung, diesen Monat wird Ihre Kombinationsfähigkeit gefragt: beim Zahlenraten, Schiffeversenken oder den vertrackten "Magic Rings" raucht mal nicht der Prozessor Ihres Rechners, sondern Ihr eigener Kopf.

Aller Anfang ist bei uns nicht etwa schwer, sondern ungemein pfiffig: Wir starten mit einem trickreichen Utility, das Ihren Bildschirm etwas durcheinanderbringt.

Bildschirmwechsel

Viele Programme laden Grafikbildschirme als Dateien von Cassette oder Diskette nach. Leute, die mit solchen Programmen arbeiten, sind oft auf der Suche nach Routinen, mit denen sich der Bildwechsel interessanter gestalten läßt. Bei unserem Programm BSCREEN wird das alte Bild Punkt für Punkt durch das neue ersetzt. Vorausgesetzt natürlich, daß sich dieses neue Bild bereits im Speicher ab der Adresse &4000 befindet.

Nach dem Start von BSCREEN.BAS wird zunächst der Code in den Speicher gepoket. Anschließend ruft ein "Call &A000 die 43 Bytes kleine Routine auf. Leider verschwendet das Programm in der abgedruckten Form bei der eigenen Softwareentwicklung eine ganze Menge Speicherplatz. Darum sollten Sie in der Zeile 31 den REMark-Befehl entfernen und das Programm erneut starten. Es wird nun die Datei BSCREEN.BIN auf dem Datenträger erzeugt. Sie kann mittels LOAD"BSCREEN.BIN einfach in eigene Programme übernommen werden.

Falls Sie sich eine eigene Bildershow zusammenstellen wollen, orientieren Sie sich bitte an folgendem kleinen Programmbeispiel:

```
10 MEMORY &3FFF:LOAD"BSCREEN.BIN"
20 LOAD"1. Bild",&4000
30 CALL &A000
40 LOAD"2. Bild",&4000
50 CALL &A000
60 ...
```

Viel Spaß beim Austesten! Übrigens: das Programm läuft auf allen CPCs – also auf dem CPC 464,664 oder 6128 einschließlich der "Plus"-Serie.

Clemens Lutz/rs

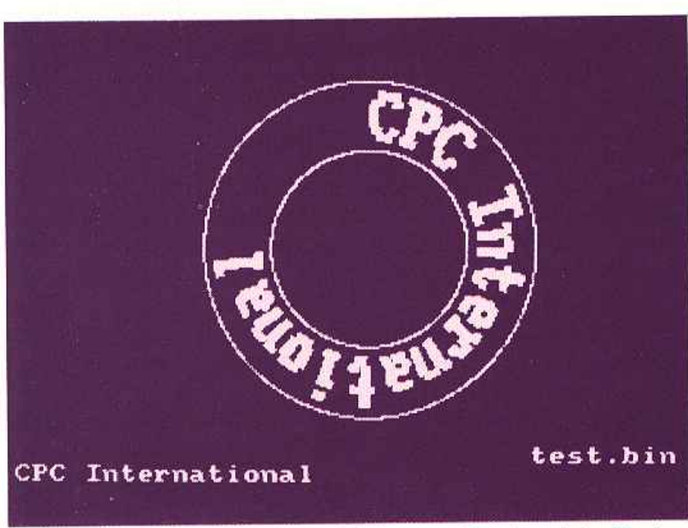
Kreisschrift

Bei diesem netten Programm für alle CPCs handelt es sich, wie in der Abbildung "Kreisschrift" gezeigt, um eine Routine, die Schriften in eine Kreisform setzt. Hierbei geben Sie lediglich den Bildschirmmodus sowie einen beliebigen Text mit bis zu 44 Buchstaben in Mode 2 ein. Im Modus 0 stehen elf Zeichen zur Verfügung, und der Modus 1 bietet ganze 22 Zeichen an. Wenn der Modus 1 angewählt und kein Text eingegeben wurde, nimmt der CPC zur Demonstration einfach den Text "+ 100 DM für 1 kByte +" als Schriftzug an.

Diesen Text, der zugegebenermaßen nicht ganz frei von werbemäßigen Hintergedanken ist, können Sie natürlich aus dem Listing eliminieren. Setzten Sie statt dessen Ihren "dummen Spruch des Tages" als Standard-Text ein.

Mit der Fertigstellung der Grafik ist der CPC je nach benutztem Modus unterschiedlich lange beschäftigt. Anschließend fragt er Sie, ob Sie die Grafik abspeichern wollen. Einer Weiterverarbeitung des computererzeugten "Button-Texts" steht nichts im Wege. Dank der Verwendung aller drei Bildschirmmodi ist diese Routine äußerst flexibel. So kann man sie dazu verwenden, schon einmal vorab ein grobes Titelbild zu erstellen, was eben eine enorme Erleichterung darstellt. Es empfiehlt sich übrigens, die Schriftbilder in Mode 1 erstellen zu lassen, da hier die beste Qualität erreicht wird und vor allem die Schrift am besten zu lesen ist. Der Modus eins stellt eine recht gute Kombination aus der Anzahl der verwendbaren Zeichen und der Qualität der entstehenden Grafik dar.

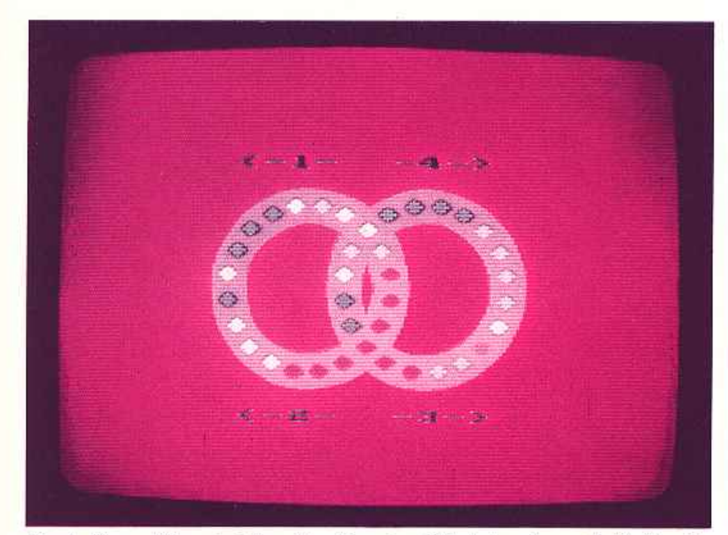
Friedrich Hinrichs/rs



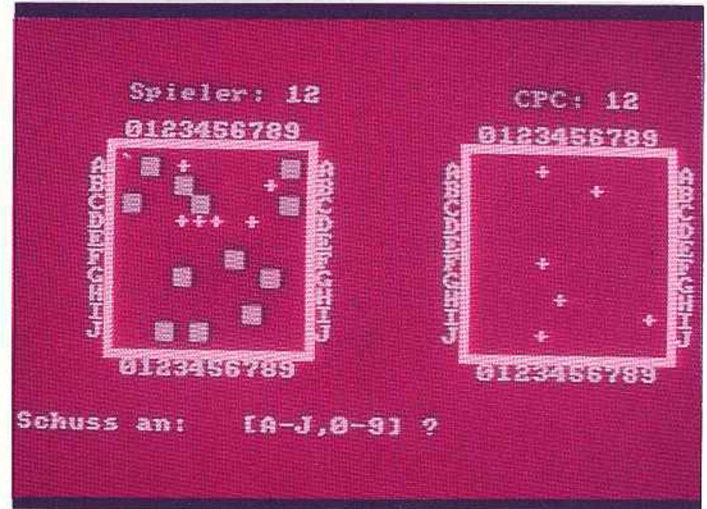
Kreisschrift: Diesmal wird die CPC International nicht auf den Kopf gestellt, oder?



Zahlenraten: Ganz so einfach wie es aussieht, ist es wirklich nicht. Spätestens nach der dritten Erhöhung wird es spannend.



Magic Rings: Hier wird Ihre Kombinationsfähigkeit gefragt. Schließen Sie die Kugeln farblich zusammenpassend aneinander.



Schiffe versenken: Das altbekannte Spiel in nur einem kByte.

Zahlenraten

Für alle, die gerne ein wenig mit Zahlen herumspielen, wurde das Programm Riddle verfaßt.

Riddle ist ein Ratespiel, bei dem sich der Computer eine Zahl ausdenkt, die der Spieler erraten muß. Geben Sie auf die Frage des Computers hin eine Zahl ein, gibt dieser Ihnen einen Hinweis, ob die gesuchte Zahl größer oder kleiner ist. Haben Sie die Zahl nach zehn Versuchen nicht erraten, ist das Spiel vorbei.

Haben Sie sie jedoch gefunden, wird das Spiel fortgesetzt. Der Zahlenbereich, aus dem der Computer sich eine Zahl aussucht, erhöht sich jedoch von zehn auf zwanzig. Bei richtigem Raten geht es dann immer so weiter: Der Zahlenbereich wird jedesmal um zehn größer. Die am Bildschirm dargestellten Punkte errechnen sich aus der Zahl der noch übrigen Versuche multipiziert mit zehn.

Ralf Möst/rs

Magic Rings

Dieses Spielprogramm simuliert das gleichnamige Spielzeug, bei welchem es darauf ankommt, durch geschicktes Drehen zweier mit Kugeln besetzter Ringe alle Kugeln einer Farbe zusammenzusetzen. Wen dies an Rubiks "Zauberwürfel" erinnert, der liegt im Prinzip richtig, aber die "Magic Rings" haben doch auch ihren eigenen Reiz.

Am Anfang erzeugt das Programm den richtigen Stand der Ringe (alle Kugeln sind in den richtigen Positionen). Dann vermischt es das Ganze um fünfzig Züge. Möchte man sich den richtigen Stand anschauen, sollte man durch Entfernen der Zeile 14 den Mischvorgang einfach weglasssen. Es werden nun die richtigen Positionen der Kugeln angezeigt.

Sollten Sie es einmal geschafft haben, die Kugeln richtig zu setzen, so seien Sie bitte nicht sauer, weil das Programm dies nicht anzeigt. Um die 1-kByte-Grenze nicht zu überschreiten, mußte darauf verzichtet werden.

Aber man kann ja selber – wie es in der 1-kByte-Rubrik eigentlich üblich ist – eine Siegerehrungs-Routine, die zum Beispiel ein Liedlein trällert, einbinden.

Petr Potuznik/rs

Schiffe versenken

Wer kennt es nicht, dieses Spiel, bei dem es darum geht, auf einem vorgegebenen Feld die Schiffe des Gegners ausfindig zu machen und zu vernichten? Generationen von Schülern haben sich damit langweilige Lateinstunden vertrieben. Das klassische Strategie-Training, gespielt auf vier Karo-Papierchen (je zwei für jeden, davon immer eins für die eigenen und eins für die gegnerischen Operationen) ist auch als das "Schachspiel des kleinen Mannes" bekannt geworden.

Diese 1-kByte-CPC-Version allerdings macht die Sache noch etwas schwieriger als gewöhnlich. Hier gibt es keine "Dreier", "Zweier" oder "Fünfer", sondern nur zwölf Schiffe, die aus je einem einzigen Element bestehen.

Also machen Sie sich daran, die Flotte Ihres Gegners zu vernichten. Übrigens: Ihr Gegner ist der Computer, der Ihnen sicherlich nichts schenkt. Tun Sie das also auch nicht, sondern zeigen Sie dem Ding, was eine Harke ist!

Petr Potuznik/rs

Mini-Vokabelpauker

Aller Anfang ist schwer. Gerade bei der Vokabellernerei kann man sich oft nur schwer aufraffen, sich doch einmal hinzusetzen und anzufangen. Wenn nun aber die Arbeit am Computer verrichtet werden kann, wird die Motivation sicherlich gleich viel höher sein. Nun – ohne Vokabeltrainer geht es nicht, und diese Programme sind in den meisten Fällen ellenlang. Dieser nicht.

Haben Sie das Programm VOKABEL.BAS eingegeben, können Sie gleich den ersten Menüpunkt anwählen und mit dem Eingeben der Vokabeln beginnen. Bevor Sie jetzt den zweiten Menüpunkt (Abfragen) wählen, speichern Sie mittels "4" zuerst einmal die Vokabeln ab. Nun können Sie sich Vokabel für Vokabel abfragen lassen. Beim nächsten Start sollten Sie jedoch nicht vergessen, die bisher eingegebenen Daten einzuladen, sonst werden diese bei erneutem Abspeichern überschrieben und sind dann verloren.

Holger Heyer/rs

```
10 'BSCREEN.BAS (1kb)                              [1249]
11 DATA A000,21,00,c0,06,dc,ed,5f,e6,7f,85 [3205]
,6f,30,15,24,20,12,fc47
12 DATA a010,7c,f6,c0,67,10,ef,21,00,40,11 [2954]
,00,c0,01,00,40,ed,a63b
13 DATA a020,b0,c9,7c,e6,7f,57,5d,1a,77,18 [2354]
,da,00,00,00,00,00,82a9
14 DATA *Ende*                                     [502]
15 adr=&A000:zeile=100                             [1435]
16 MEMORY adr-1 ,                                  [160]
17 READ d$                                         [295]
18 IF d$="*Ende*" THEN 31                          [907]
19 pr=0                                            [117]
20 FOR i=1 TO 16                                   [317]
21 READ a$:a=VAL("&"+a$)                           [883]
22 POKE adr,a:adr=adr+1                            [717]
23 pr=pr*2:IF pr>65535 THEN pr=pr-65535            [2485]
24 pr=UNT(pr)XOR a:IF pr<0 THEN pr=pr+6553 [1890]
6
25 NEXT i                                          [375]
26 READ pr$:pr2=VAL("&"+pr$):IF pr2<0 THEN [2610]
 pr2=pr2+65536
27 IF pr<>pr2 THEN 30                              [847]
28 zeile=zeile+10                                  [568]
29 GOTO 17                                         [397]
30 PRINT"Pruefsummenfehler in Zeile";zeile [4530]
:STOP
31 'SAVE"BSCreen.bin",b,&A000,&2B:END              [3184]
32 CALL &A000                                      [637]
```

```
10 'KREISCH.BAS (1kb)                              [973]
11 INK 0,0:INK 1,26:BORDER 0:MODE 1:DEG:SY [4606]
MBOL AFTER 32:SYMBOL 96,102,0,102,102,102,
102,62,0:a$="+ 100 DM f'r 1 kByte +"
12 INPUT"MODUS ",m:CLS:PRINT"GEBEN SIE DIE [9461]
 SCHRIFTZEICHEN EIN,DIE IN DEN KREIS GESET
ZT WERDEN SOLLEN          ( max.";2^m*11;")
":INPUT" ",zeich$:IF zeich$="" AND m=1 THE
N zeich$=a$
13 MODE m:st=0.75/2^m:LOCATE 1,25:PRINT ze [1827]
ich$
14 FOR a=1 TO 360 STEP st:PLOT 320+85*SIN( [6183]
a),200+85*COS(a):PLOT 320+145*SIN(a),200+1
45*COS(a):NEXT
15 FOR x=0 TO 360 STEP st:FOR y=0 TO 16 ST [2349]
EP st
16 IF TEST(x,y) THEN PLOT 320+(y*3+90)*SIN [3416]
(x),200+(y*3+90)*COS(x)
17 NEXT y,x                                        [397]
18 LOCATE 1,24:INPUT "ALS GRAPHIK ABSPEICH [5284]
ERN (J/N)",jn$:IF jn$="J" OR jn$="j" THEN
GOTO 20
19 RUN                                             [243]
20 LOCATE 1,24:INPUT"NAME DER GRAPHIK              [6815]
        ",name$:LOCATE 1,24:PRINT"
                        ":SAVE name$,b,49152,

16384:RUN
```

```
10 'RIDDLE.BAS (1kb)                               [608]
11 MODE 1:p=10:f=10:v=11                           [990]
12 WINDOW#1,15,24,9,11:WINDOW#2,14,25,14,1 [2300]
6
13 z=INT(RND*f)+1                                  [938]
14 FOR x=1 TO 10:v=v-1:LOCATE 1,1:PRINT v" [8931]
Versuche ":LOCATE 25,1:PRINT "Punkte:";p:L
OCATE 13,23:PRINT"Bereich 0 -";f
15 MOVE 180,120:DRAW 180,290:MOVE 425,120: [5943]
DRAW 425,290:MOVE 180,290:DRAW 425,290:MOV
E 180,120:DRAW 425,120
16 PAPER#2,3:PEN#2,0:CLS#2:PAPER#1,3:PEN#1 [5119]
,0:CLS#1:PRINT#1:INPUT#1," ZAHL";zahl
17 IF zahl>z THEN PRINT#2:PRINT#2,"  Klein [9705]
er "ELSE IF zahl<z THEN LOCATE 15,15:PRINT
#2:PRINT#2,"  Groesser"ELSE IF zahl=z THEN
 LOCATE 15,15:PRINT#2:PRINT#2,"  GEWONNEN"
:x=10
18 IF z<>zahl AND x=10 THEN LOCATE 12,15:P [12380]
RINT"G A M E  O V E R":p=0:f=0:CALL &BB18:
PRINT:INPUT"Noch einmal (j/n) ";a$:IF a$="
j" THEN GOTO 11 ELSE END:CLS:PRINT"Auf Wie
dersehen !":IF z=zahl AND x<11 THEN p=p*v
ELSE p=p:IF v=0 THEN v=11
19 IF z=zahl AND x<11 THEN p=p*v ELSE p=p: [4287]
IF v=0 THEN v=11
20 NEXT:CALL &BB18:CLS:f=f+10:v=11:GOTO 13 [2149]
```

```
10 'RINGS.BAS (1kb)                                [1097]
11 DIM a(2,19):r(1)=238:r(2)=380:DEG:BORDE [11134]
R 0:FOR i=0 TO 6:READ b:INK i,b:NEXT:SYMBO
L 255,16,56,56,124,124,56,56,16:MODE 0:LOC
ATE 6,4:PRINT"<-1-  -4->":LOCATE 6,23:PRIN
T"<-2-  -3->":TAG
12 FOR i=1 TO 2:FOR r=80 TO 120 STEP 40:MO [10624]
VE r(i)+14,192+r:FOR x=10 TO 360 STEP 10:D
RAW r*SIN(x)+r(i)+14,r*COS(x)+192,6:NEXT x
,r:MOVE r(i),99:FILL 6:NEXT i:MOVE 280,200
:FILL 6:GRAPHICS PAPER 6
13 FOR i=1 TO 2:FOR x=0 TO 18:a(i,(x+28+10 [3812]
*(i=2))MOD 20)=2*i-(x>9):NEXT:NEXT:a(1,7)=
5:a(2,17)=3
14 FOR z=0 TO 50:b=1+INT(3.99*RND):GOSUB 1 [3524]
9:NEXT:' MISCHEN
15 i=1:GOSUB 18:i=2:GOSUB 18                       [1654]
16 b=VAL(INKEY$):IF b<1 OR b>4 THEN 16             [3342]
17 GOSUB 19:GOSUB 18:GOTO 16                       [2323]
18 FOR x=0 TO 19:GRAPHICS PEN a(i,(x+s(i)) [8354]
MOD 20):MOVE 100*SIN(x*18+9)+r(i),100*COS(
x*18+9)+200:PRINT CHR$(255);:NEXT:RETURN
19 i=1-(b>2):s(i)=(s(i)+19+2*(b MOD 2))MOD [10057]
 20:j=(17+s(2))MOD 20:k=(2+s(1))MOD 20:m=(
12+s(2))MOD 20:n=(7+s(1))MOD 20:IF i=1 THE
N a(2,j)=a(1,k):a(2,m)=a(1,n) ELSE a(1,k)=
a(2,j):a(1,n)=a(2,m)
20 RETURN:DATA 0,2,6,26,11,12,1                    [2058]
```

```
10 'SCHIFF.BAS (1kb)                               [1289]
11 n=12:m=n:DIM p$(n),s(100),t(100):s(100) [9126]
=1:FOR i=1 TO 3:p$(i)=CHR$(15)+CHR$(i):NEX
T:MODE 1:LOCATE 7,2:PRINT p$(3);"Spieler:"
;SPACE$(14);"CPC:";p$(1)
12 FOR i=1 TO 2:WINDOW #i,20*i-15,40,4,25: [8179]
PRINT #i,"  0123456789":PRINT #i," ";p$(2)
;CHR$(135);STRING$(10,131);CHR$(139):FOR j
=1 TO 10:PRINT #i,p$(1);CHR$(64+j);p$(2);C
HR$(133);SPACE$(10);CHR$(138);p$(1);CHR$(6
4+j)
13 NEXT:PRINT #i," ";p$(2);CHR$(141);STRIN [4693]
G$(10,140);CHR$(142):PRINT #i,p$(1);"  012
3456789":NEXT
14 LOCATE 1,20:PRINT p$(2);"Position einge [16855]
ben: [A-J,0-9]";p$(1):FOR i=1 TO n:LOCATE
1,22:PRINT USING "##";i;:PRINT ".Boot:":x=
100:WHILE s(x)=1:LOCATE 10,22:PRINT CHR$(2
0);:GOSUB 19:x=10*b+a:WEND:s(x)=1:LOCATE 7
+a,6+b:PRINT p$(3);CHR$(143);p$(1)
15 LOCATE 15,2:PRINT i:a=INT(RND*99):WHILE [7618]
 t(a):a=(a+1)MOD 100:WEND:t(a)=1:LOCATE 33
,2:PRINT i:NEXT:LOCATE 1,20:PRINT CHR$(20)
;p$(1);"Schuss an:   [A-J,0-9] ?"
16 LOCATE 27,20:PRINT CHR$(20);:GOSUB 19:j [7225]
=10*b+a:LOCATE 27+a,6+b:IF t(j)<>1 THEN PR
INT "+" ELSE t(j)=0:m=m-1:GOSUB 20:LOCATE
33,2:PRINT m:IF m=0 THEN 18
17 a=INT(RND*99.99):WHILE s(a)>1:a=(a+1)MO [10838]
D 100:WEND:LOCATE 7+a MOD 10,6+INT(a/10):I
F s(a)<>1 THEN PRINT "+":s(a)=3:GOTO 16 EL
SE s(a)=2:n=n-1:GOSUB 20:LOCATE 15,2:PRINT
 n:IF n>0 THEN 16
18 LOCATE 1,20:PRINT p$(3);CHR$(20);"ENDE. [7961]
":FOR i=1 TO 100: WHILE t(i)=1:LOCATE 27+i
 MOD 10,6+INT(i/10):PRINT CHR$(143):t(i)=4
:WEND:NEXT:END
19 a$="":WHILE a$<"A" OR a$>"J":a$=UPPER$( [7300]
INKEY$):WEND:PRINT p$(2);a$;" -";:b=ASC(a$
)-65:a=0:WHILE (a=0)*(a$<>"0")=1 OR a>9:a$
=INKEY$:a=VAL(a$):WEND:PRINT a;:RETURN
20 PRINT p$(1);CHR$(206);p$(2):SOUND 1,200 [3418]
,20,,,,15:RETURN
```

```
10 'VOKABEL.BAS (1kb)                              [803]
11 MODE 2:LOCATE 28,1:PRINT"VOKABELTRAINER [6301]
":LOCATE 28,2:PRINT"=============="
12 RESTORE 12:FOR a=4 TO 12 STEP 2:READ a$ [8943]
:LOCATE 29,a:PRINT a$:NEXT:DATA 1=EINGEBEN
,2=ABFRAGEN,3=LADEN,4=SPEICHERN,5=ENDE
13 a$=INKEY$:IF a$="" THEN 13 ELSE IF a$=" [7750]
2" THEN LOCATE 34,18:PRINT"[TASTE]":WHILE
INKEY$="":v=v+1:IF v>va THEN v=1
14 IF a$="2" THEN WEND:LOCATE 34,18:PRINT" [14305]
            ":LOCATE 1,20:PRINT v$(v):LOCAT
E 60,20:INPUT":",an$:IF an$<>ve$(v) THEN L
OCATE 40,24:PRINT"Falsch":WHILE INKEY$="":
WEND:f=f+1:GOTO 11 ELSE LOCATE 40,24:PRINT
"RICHTIG":WHILE INKEY$="":WEND:r=r+1:GOTO
11
15 IF a$="1" THEN LOCATE 1,20:va=va+1:INPU [5898]
T"DEUTSCH:",v$(va):LOCATE 50,20:INPUT"ENGL
ISCH:",ve$(va):GOTO 11
16 IF a$="3" THEN en$="*.DAT":öDIR,@en$:IN [9724]
PUT"FILENAME:",n$:OPENIN n$+".dat":va=0:WH
ILE NOT EOF:va=va+1:INPUT#9,v$(va),ve$(va)
:WEND:CLOSEIN:GOTO 11
17 IF a$="4" THEN INPUT"FILENAME:",n$:OPEN [8704]
OUT n$+".dat":WHILE anz<va:anz=anz+1:WRITE
#9,v$(anz),ve$(anz):WEND:CLOSEOUT:GOTO 11
18 IF a$="5" THEN CLS:PRINT"Richtig:",r:PR [3463]
INT"Falsch:",f
```



Gamers Message

Tips und Tricks zu beliebten Spielen

Jeder hat es schon einmal erlebt: Man ist von einem Spiel wahnsinnig begeistert, aber man ist an einer Stelle, wo es einfach nicht weitergeht. Auch neu erworbene Spiele werden manchmal zum Problem, weil man einfach keinen Einstieg findet. Wir versuchen, Ihnen zu helfen.

Zu Beginn gleich einige Tips zu den in diesem Heft vorgestellten Spielen. Los geht's mit **Dick Tracy**. An manchen Stellen kommt man ohne eine Schußwaffe nicht weiter. Normale Gangster können mit einem Faustschlag beseitigt

Dick Tracy

werden (wenn man sich duckt, ist man sicherer). An den Schüssen aus Fenstern kommt man vorbei, indem man schnell läuft und gegebenenfalls springt. Auf diese Weise kann man sich die Munition aufsparen. Die Leute, die Bomben aus dem Fenster werfen, müssen aber ebenso abgeschossen werden wie die bewaffneten Feinde. Um nicht gleich getroffen zu werden, wenn man weitergeht, ist es am günstigsten, nur sehr kurze Strecken zu gehen und dann stehenzubleiben. Wenn der bewaffnete Gegner noch am äußersten rechten Bildschirmrand ist, rührt er sich nicht, kann aber in aller Ruhe erschossen werden. Wenn man sich an diese Technik hält, kann man relativ sicher durch die Stadt marschieren.

Bei **Switchblade** neigt man dazu, schnell nach rechts zu gehen und sich ins Getümmel zu stürzen. Besser ist der Weg zu dem Gitterkäfig, der den Eingang zu dem unterirdischen Labyrinth markiert. Hier beseitigt man am besten gleich alle Monster, läßt sich aber dann nicht fallen. Statt dessen sollte man umdrehen, nach links weitergehen und auch hier den Weg säubern. Ganz links angekommen, findet man eine kleine Höhle, in der man eine Schußwaffe bekommt. Wieder nach rechts, und im eigentlichen Höhlensystem angekommen, ist man gleich bewaffnet.

Erst einmal bewaffnen

Wenn man fällt, liegt die Spielfigur benommen am Boden und hat fast keine Energie mehr. Geraten Sie jetzt nur nicht in Panik. Warten Sie, bis die Figur wieder steht. Nun zerschlagen Sie links die Steine, und Sie erhalten wieder Energie. Generell sollte man sich Zeit lassen. Alle Steine, die einen Sprung haben, können zerschlagen werden. So findet man oft Extras oder Wege, die Gefahren geschickt umgehen.

Taucht ein Feind auf, wartet man am besten, ob er sich auf den Spieler zubewegt. Wenn er das tut, einfach warten und ihn herankommen lassen. Dadurch vermeidet man, zwischen zwei Gegnern eingeklemmt zu werden. Auch

Switchblade

sonst ist Hektik tödlich. Bei den Spezial-Gegnern ist es besser, sich rechtzeitig zurückzuziehen und den nächsten Schuß abzuwarten, als von diesem getroffen zu werden.

In jedem Spiel den gleichen Fehler zu machen, ist ärgerlich. Wer sich also nicht merken kann, wo welche Gefahren und Extras warten, sollte sich eine Karte zeichnen oder Notizen machen. Sonst hat man keine Chance, und Harvok (oder der Programmierer) lacht sich ins Fäustchen.

Wenn sich der Norden und der Süden in **North & South** gegenüberstehen, hat nur derjenige eine Chance, der eine funktionierende Eisenbahnlinie kontrolliert. Diese sollte also das erste und wichtigste Augenmerk sein. Das Zusammenlegen von Armeen ist nur sinnvoll, wenn die bevorstehende Schlacht

North & South

extrem wichtig ist. Ansonsten kommt man auch gut mit schwachen Truppen aus. Das sollte man vor allem beachten, wenn es nur um Landgewinn geht, und keine gegnerischen Einheiten auf dem angegriffenen Land stehen.

In den Schlachtfeldszenen sind die Kanonen sehr nützlich, doch ihre Munition ist begrenzt. Man muß also versuchen, die zur Verfügung stehenden Schüsse möglichst effektiv zu nutzen. Am besten tut man dies, indem man auf eine Einheit zielt, die der Gegner gerade nicht im Auge hat.

Der Schlüssel zum Erfolg im ersten Teil von **Batman** ist ein gutes Gedächtnis – oder eine Karte. Ohne dies hat man kaum eine Chance. Sehr zu empfehlen ist es auch, sich sehr genau mit der Verwendung des Seils vertraut zu machen. Diese Tips werden auch im fünften Teil nützlich sein.

Im zweiten Teil muß man vor allem herausfinden, in welchem Moment man das Lenkrad herumreißen muß,

um richtig abzubiegen. Der dritte Level läßt sich durch bloßes Ausprobieren lösen. Wenn es in der vierten Runde darum geht, die Ballons abzuschneiden, ist es gut, wann immer möglich an den rechten Bildschirmrand zu fliegen. Wenn man dann von oben nach unten (oder umgekehrt) fliegen muß, kann

Batman

man Zeit gewinnen, wenn man gleichzeitig schräg nach hinten drückt.
Die Anleitung **Vendettas** enthält allerlei Hinweise. Doch eines kann sie dem Spieler nicht abnehmen: herumzuprobieren, an welche Ecke des Objekts die Spielfigur plaziert werden muß, damit die gewünschte Aktion ausgeführt werden kann. Außerdem: Wenn fünf Schreibtische leer sind, heißt das noch lange nicht, daß der sechste auch leer ist. Noch ein Tip: Das Spiel ist nicht nur etwas für Autofahrer, sondern auch etwas für Computerfreunde. Denken Sie daran!
Bei dem Rollenspiel-Klassiker **The Bard's Tale** bereitet vor allem der Anfang Schwierigkeiten. Zuerst ist da einmal die Auswahl der Charaktere. Weil sie im weiteren Spielverlauf extrem wichtig werden, sollte man mindestens zwei magisch talentierte Charaktere mitnehmen. Auch ein Barde ist nötig. Anfangs mag man zwar glauben, ohne ihn auskommen zu können, aber im

Bard's Tale

weiteren Spielverlauf wird man mit einem Problem konfrontiert werden, das man ohne einen Barden nicht lösen kann. Verzichten kann man hingegen auf einen Dieb. Das Entschärfen der Fallen wird dann mit einem *Trap-Zap*-Spruch vorgenommen.

Man sollte seine Kraft nie überschätzen

Ist man etwas tiefer ins Spiel vorgedrungen, sind Gruppenmitglieder, die sich aufs Zuschlagen spezialisiert haben, nicht mehr besonders nützlich. Wer allerdings anfangs keine davon hat, wird erst gar nicht so weit kommen ... Noch eine Anmerkung zu dem Barden der vorbereiteten Party: Er hat ein *Fire Horn*, ein äußerst nützliches Gerät, das es nirgends zu kaufen gibt. Man sollte es aber nur sparsam verwenden, da es nicht ewig hält.

Da man zu Beginn noch schwach ist, sollte man sich nicht zu weit von der *Adventurer's Guild* entfernen und nur gegen kleine Gruppen schwacher Gegner kämpfen. Ist die angreifende Monstergruppe zu stark, sollte man versuchen davonzulaufen. Weil dies nicht immer funktioniert, ist es ratsam, nach jedem gewonnenen Kampf das Spiel zu speichern. Als nächstes sollte man daran gehen, sich eine Orientierung in der Stadt zu verschaffen. Weil der Computer – anders als in der Anleitung angegeben – die Namen der Straßen nicht anzeigt, ist dies gar nicht so einfach. Hat man genügend Erfahrung gesammelt, kann man sich in die Dungeons wagen, wo mehr Gefahren, aber auch mehr Schätze und mehr Ruhm warten ...
Weiter geht's mit einem anderen Rollenspiel, nämlich **Bloodwych**. Die Anleitung ist zwar dank ihrer Übersetzung für einige Heiterkeit gut, doch verschweigt sie leider einen wichtigen Punkt: Die Spielstände müssen auf der

Bloodwych

Originaldiskette abgespeichert werden. Benutzt man eine andere Diskette, sind die Effekte zwar interessant, erleichtern das Spiel aber nicht gerade. Wer mißtrauisch ist, sollte vielleicht vorher eine Sicherheitskopie der Originaldiskette anfertigen.
Obwohl der Anfang nicht so schwierig ist wie bei *The Bard's Tale*, sollte man doch auch hier häufig den Spielstand speichern und unbedingt sofort damit beginnen, eine Karte zu zeichnen! Wichtig ist zunächst, sich an das anfangs etwas kompliziert erscheinende Menüsystem zu gewöhnen. Die ersten Gegner werden dann nicht lange auf sich warten lassen. Wenn man darauf achtet, nicht zwischen zwei Monstergruppen zu geraten, sollten diese jedoch kein allzu großes Problem darstellen. Die Farbe der Türen hat übrigens etwas mit der Farbe des Schlüssels zu tun, der sie öffnet. Und einige Wände verschwinden auf Knopfdruck ...

Wer bei **Sim City** Schwierigkeiten hat, mit seinem Geld auszukommen, kann

Sim City

es wie unsere Politiker machen und die Bevölkerung an der Nase herumführen. Die Steuern werden im Dezember auf zwanzig Prozent erhöht und fließen dann in die Stadtkasse. Bevor jedoch die Einwohner reagieren und auswandern können, wird der Steuersatz im Januar wieder auf null Prozent gesetzt, um neue Siedler anzulocken. Bis sich dann dasselbe im Dezember wiederholt.

Vom Spiel zur Wirklichkeit

The Sentinel ist ein recht außergewöhnliches Spiel. Das äußert sich leider auch darin, daß man nach der Lektüre der Anleitung etwa genauso schlau ist wie zuvor. Zwar steht alles drin,





CPC-Anwendungen

Die bewährten und praxiserprobten CPC-Anwendungsprogramme aus dem DMV-Verlag bringen Ihre CPC-Hardware richtig in Schwung: Sie erzielen hervorragende Ergebnisse in kürzester Zeit

ConText CPC

Professionelle Textverarbeitung

für alle CPC

Lassen Sie Ihren CPC nicht verstauben! Mit ConText CPC können Sie ein Anwendungsprogramm erwerben, das für Textverarbeitung eine überzeugende Leistung bei einfacher Bedienung bietet. Die ausführliche deutsche Anleitung macht auch Computerlaien nach kürzester Zeit eine Textverarbeitung per Computer möglich!

Mit ConText CPC macht das Schreiben richtig Spaß:

● Einfügen, Fließtext, Blockformatierung und Kopieren auf Tastendruck ● 25 KByte Textspeicher, ausreichend für mindestens fünf DIN-A4-Seiten ● Mehrspaltige Texterstellung und Kopieren für besondere Aufgaben ● Textblöcke von Diskette oder Kassette jederzeit im Text zu plazieren ● Drucken im Hintergrund, während Sie weiter an Ihren Texten arbeiten ● Echtbild-Darstellung der Schriftarten »Vergrößert« und »Unterstrichen« ● In das Programm ConText CPC integrierter Taschenrechner und Kalender ● Variable Druckeranpassung mit Einstellmöglichkeit in Tabellenform

ConText CPC
Hervorragend in Preis und Leistung DM **59,-** *

CopyShop

Das universelle Hardcopy-Programm

für alle CPC

Wollen Sie Ihre traumhaft schöne Fraktalgrafik im Freundeskreis zeigen, oder brauchen Sie zu Ihrem selbstgeschriebenen Programm einen Ausdruck? Mit CopyShop bringen Sie alle Bildschirmansichten zu Papier, auch mit den exotischsten Druckern. Ein Show-Modus zeigt Ihnen die Hardcopy vor dem endgültigen Ausdruck auf dem Bildschirm!

Mit CopyShop Hardcopies immer und überall:

● Vier Formate in allen drei CPC-Modi mit automatischer Erkennung ● In das Programm integrierter Grafikeditor mit Füllfunktion ● 32 Farbraster wählbar über komfortable Pull-down-Menüs ● Beliebige Ausschnittsvergrößerungen mit Invertierungsfunktion ● Drucker-Anpaßmenü für alle Epson- und Seikosha-Drucker ● Anpassung auch an gedrehte Bitbild-Bytes, z.B. an NEC P2 ● Freezer auch für Hardcopies aus laufenden Basic-Programmen ● selbstrelozierende Hardcopy-Routinen für eigene Programme

CopyShop
Das ultimative Hardcopy-Programm DM **49,-** *

Turbo-DATA-CPC

Volldampf in der Dateiverwaltung

Brauchen Sie ein neues Adreßbuch oder Telefonverzeichnis? Müssen Sie Ihre Sammlungen von Briefmarken, Schallplatten, Dias oder anderen Schätzen sortieren? Alles kein Problem mit Turbo-DATA-CPC, der universellen Dateiverwaltung für alle CPC mit Diskettenlaufwerk, mit der Sie Überblick und Ordnung in Ihre Daten bringen.

Datenverwaltung komplett mit Turbo-DATA-CPC:

- Universell durch veränderbare Ein- und Ausgabemasken
- Blitzschnelle Suchfunktionen durch Indexfelder
- Dateikapazität bis maximal 80 verschiedene Felder
- Bis zu 19 Felder gleichzeitig auf dem Bildschirm sichtbar
- Besonders schnelle und umfangreiche Sortierfunktion
- Eigene Formatierroutine mit extrem hoher Kapazität
- Zweiteilung des Bildschirms in Status- und Arbeitsfeld
- Komplette Druckroutinen, auch für Etikettendruck

Turbo-DATA-CPC
Immer die richtige Wahl DM **69,-** *

CPC Special Offer 2

Vier Programme im Paket-Sonderpreis

Ob Bildschirmgrafiken, Sprite-Animation, Diskettenmonitor oder eine Diskettenverwaltung – im Special Offer 2 für alle CPC ist für jeden etwas dabei:

DISKSORT-STAR
Leistungsstarke Diskettenverwaltung mit hohem Bedienungskomfort und integriertem Diskettenmanager. Verwaltung, Archivierung, Katalogisierung Ihrer Disketten und Ausdruck der Daten - mit Disksort-Star kein Problem.

DESIGNER-STAR
Grafikprogramm zum komfortablen Erstellen von Bildschirmgrafiken. Zur Bedienung ist ein Joystick oder eine Maus nicht notwendig. Auf Tastendruck werden Sie mit Hilfsmenüs durch das Programm geführt.

CREATOR-STAR
Erstellen Sie eigene Trickfilme auf Ihrem CPC - mit einem Kulissen- und Sprite-Designer sowie beliebigen Laufschriften. In einer eigenen Programmiersprache mit Editor und Compiler können Sprites verbunden und Kulissen übereinandergelegt werden.

STAR-MON
Professionelle CPC-Programmierung mit Assembler, Disassembler und Monitor. Das System besitzt einen eigenen Diskettenmonitor und Editor, eine Trace-Funktion, kann Breakpoints setzen und Bankswitching vornehmen.

CPC Special Offer 2
Für CPC-Profis DM **59,-** *



Das Software-Experiment

Spielend den CPC beherrschen lernen

Nehmen Sie teil am Abenteuer "Computer"! In zehn lebendig und leicht verständlich geschilderten Reisen in die Welt der Computergrafik, Simulation und künstlichen Intelligenz erarbeitet der Autor mit Ihnen zusammen insgesamt 17 interessante Programme, die Sie auf Diskette erhalten.

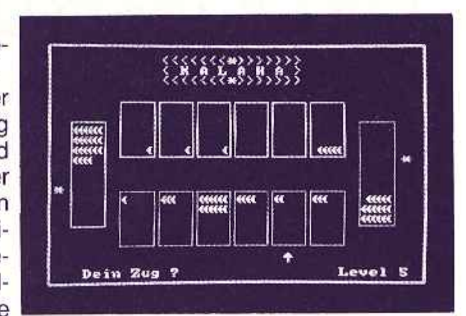
Das umfangreiche, 180seitige Handbuch vermittelt Ihnen auf anschauliche und unterhaltsame Weise, wie aus Algorithmen Datenstrukturen und letztendlich Software entstehen: von der Idee zum Programm - der ideale Kursus zum Einstieg in das Computerwissen, für Hobby und Schulunterricht. Sehen Sie selbst, zu welchen phantastischen Grafiken und verblüffenden Intelligenzleistungen der CPC bei raffinierter Programmierung fähig ist!

Wordmaster
Sie raten ein Wort, das sich der Computer ausgedacht hat. Dann rät der Rechner ein Wort, das Sie sich ausgedacht haben. Wer braucht weniger Versuche? Sie werden sich wundern...

Komplexe Grafik
Sie lernen die mathematischen Grundlagen von Fraktalgrafiken und erarbeiten einen kompletten Fraktalgenerator zur Erzeugung von "Apfelmännchen"-Bildern.

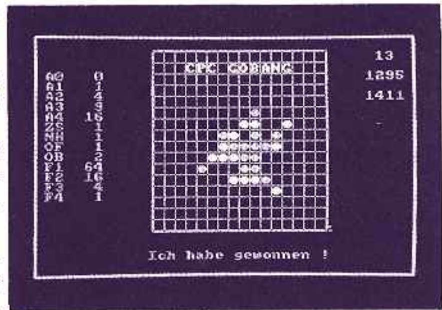
Das Spiel des Lebens
Nach natürlichen Prinzipien des Wachstums erzeugt der Computer phantastische, kristallförmige Farbstrukturen. Ändern Sie die Regeln und Voraussetzungen, und sehen Sie, was für neue, noch nie dagewesene Formen sich ergeben.

Mit roher Rechengewalt
Mit dem "Brute-Force"-Algorithmus darf der CPC so richtig loslegen und spielt mit roher Rechengewalt in dem afrikanischen Strategiespiel "Kalaha" alle Gegner an die Wand - oder können Sie ihn trotzdem schlagen?



Entwicklungshilfe
Eine gut ausgestaltete Simulation versetzt Sie in die Lage eines Entwicklungshelfers, der eine Hungersnot abzuwenden versucht. Aber Ihre Mittel sind begrenzt...

Der Computer als Stratege
Ein raffiniertes System von Spielzugbewertungen macht den CPC zu einem fast unschlagbaren Gegner im Gobang-Spiel, der Ihnen anfangs saftige Niederlagen bescheren wird. Keine Angst - das Programm ist zu schlagen - aber wie, das müssen Sie selbst herausfinden.



Der Computer lernt
Erleben Sie ein Paradebeispiel für "Künstliche Intelligenz": Das Programm "Minischach" ist lernfähig und wird mit jeder Partie ein bißchen besser.

Das Ökologie-Experiment
Der Computer zeigt Ihnen in animierter Farbgrafik einen simulierten Lebensraum mit Weideland, Schafen und Wölfen. Erforschen Sie die ökologischen Gesetzmäßigkeiten, und schaffen Sie ein stabiles, natürliches Gleichgewicht.

Wordketten
Mit einem raffinierten Algorithmus löst der CPC Wortketten-Rätsel. Oder er erfindet neue Rätsel - ganz wie Sie wollen.

Pascal läßt grüßen
Ein erstaunliches Programm enthüllt verborgene Strukturen im sogenannten "Pascalschen Dreieck". Die grafische Darstellung ergibt traumhaft schöne und farbige Musterbilder.



Das Software-Experiment

Siebzehn verschiedene, lauffähige Programme
180 Seiten Anleitung und Erklärung
Diskette nur DM **39,-***

CPC-Spiele

Zu unglaublichen Preisen!

Faszination in 3D

Zwei Super-Aktion-Spiele inklusive 3D-Brille

3D-Light Cycle

Das bekannte und beliebte TRON-Spiel für zwei Personen jetzt wie im Film! Erleben Sie dieses spannende und abwechslungsreiche Aktionsspiel jetzt in einer neuen Dimension. Verblüffende und noch nie dagewesene 3D-Effekte zusammen mit einem professionellen Sound lassen Sie in eine völlig neue CPC-Welt versinken. Diesen Super-Hit muß man live gesehen haben.

3D-Labyrinth

Das beste Labyrinthspiel jetzt in Super-3D-Qualität. Auf der Suche nach Hinweisen zum Passwort, das den mächtigen Zentralcomputer lahmlegen kann, lauern tausend Gefahren auf Sie. Räumlich perfekte 3D-Darstellung, extrem schneller Grafikaufbau und viele Überraschungen garantieren eine völlig neue Art von Spielvergnügen.

Für alle CPC 464, 664, 6128 mit Farbmonitor. Das Spiel 3D-Light Cycle kann in 2D-Darstellung auch auf Grünmonitor gespielt werden.

DM 39,-*

Know CPC

Interessantes Frage- und Antwortspiel für die ganze Familie

DM 29,-*

StarTest

Aktion-Adventure für alle CPCs

DM 19,-*

Fantastic Four

Vier Superprogramme zum Minipreis

Cockaigne - Weltraum-Aktion
Terranaut I - Science-Fiction
Fruits - Geschicklichkeit
Terranaut II - Textadventure

DM 29,-*

Special Offers III

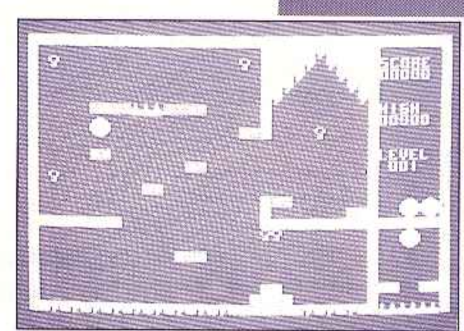
Neun interessante Spiele für alle CPCs
2 Disketten,

DM 39,-*

Game Box 3

Für jeden das Richtige - vier ausgesuchte Top-Spiele für CPC

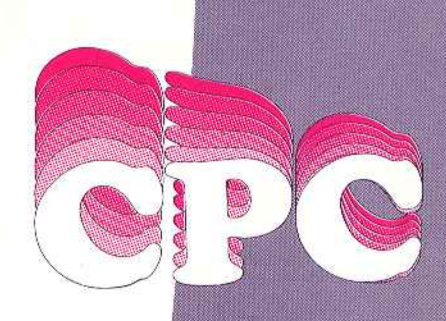
Alphajet - Weltraum-Aktionsspiel mit außergewöhnlicher Grafik und tollen Effekten



Telefomania - Interessantes Science-Fiction-Spiel mit völlig neuer Spielidee

Kampf den Insekten - Erlebnisreiches Geschicklichkeitsspiel mit Überraschungen

Funbouncer - schnelles und farbenfrohes Aktions-Spiel mit eigenem Bild-Konstruktions-Editor



DM 29,-*

CPC Joyce Preis-Offensive



Die Super-Serie



Das Beste von RE-Software: Ausgewählte und geprüfte Qualitäts-Software für CPC

Jetzt exklusiv zum Lächel-Preis beim DMV

Wie wär's denn damit:

The Player's Dream Vol. 3

Eine Sammlung besonders attraktiver Spiele, professionell programmiert und mit detailreicher Grafik ausgestattet. **Best.-Nr. 194, DM 19,-***

Ritter Kunibert: Eines der witzigsten Reaktionsspiele für den CPC, tempogeladen und mit detailreicher Farbgrafik. Verhelfen Sie Kunibert dazu, Sprung für Sprung die Burg seines Erzfeinds zu stürmen. Vorsicht: dieses Spiel macht süchtig!

Panik im Labyrinth: Etwas für Freunde gepflegter Hektik. Mitten in einem Riesenlabyrinth mit herumzappelnden und umherwischenden Energiefressern erwacht Ihre Spielfigur. Und jetzt gilt es, ein knappes Dutzend Schlösser zu finden. Das Spiel besticht durch blitzschnelle Bildwechsel und seidenweiche Joystick-Steuerung.

Bulldozer: Eine pfiffige Mischung aus "Pac-Man" und "Soko-Ban", mit liebevoll gestalteter Character-Grafik.

Froggie: "Tierisches" Jump'n-Run-Spiel mit Horizontal-Scrolling. Für 1 oder 2 Spieler simultan.

Soft-Ball: Ein Flipper mit -zig verschiedenen, zockigen Geräuschen — für alle, denen Arzt oder Ehefrau den Besuch der Kneipe an der Ecke verboten haben.

Dow Jones: Ein grafisch aufgepepptes Börsenspiel im EDV-Milieu. Kaufen Sie Atari, IBM oder Amstrad — machen Sie Millionen oder verlieren Sie Haus und Hof.

Persönlichkeitstest: Optisch gelungene "Selbstanalyse" für den kleinen Psychologen. Nicht ganz ernstgemeint, macht aber Spaß.

Skat: Eines der raren Skatprogramme. Fehlen der zweite und dritte Mann für eine zünftige Partie? Dieses Programm hilft aus. Aufwendige Farbgrafik.

Memotron: Die schönste "Memory"-Version, die es je auf einem 8-Bit-Computer gegeben hat. Sie spielen simultan gegen den Computer oder einen menschlichen Gegner und können das gegnerische Spiel durch Ihre eigenen Spielzüge manipulieren. Gutes Gedächtnis und Reaktion werden so gleichermaßen trainiert.

Q-Bert 2: Neue Version des liebenswerten klassischen Perspektiv-Hüpfspiels. Viel Feind, viel Ehr.

The Player's Dream Vol. 6

Ein Paket besonders anspruchsvoller Spielprogramme, die nie langweilig werden — mal flott und kernig, mal tüftelintensiv. **Best.-Nr. 195, DM 19,-***

Soko-Ban Constructor: Eine Herausforderung für Ihre kleinen grauen Zellen. Finden Sie die richtige Vorgehensweise für jedes Spiellevel. Mit Bildschirm-Editor zum Gestalten und Abspeichern eigener Levels. Fünf Levels werden mitgeliefert. Achtung: absolutes Suchtspiel, der Effekt ist "Lemmings"-ähnlich!

Air Traffic Control: Koordinieren Sie den Luftverkehr zwischen acht europäischen Flughäfen! Verhindern Sie Zusammenstöße, indem Sie den Piloten die richtigen Informationen geben. Aufregende Simulation im Grafikmode 2.

Trouble in Space: Strategie und Tempo in einem faszinierenden Spiel vereint. Hier wird nicht geballert, sondern mit kühlem Kopf eine Bahn für einen Space-Scooter gebaut. Sehr hohe Spielmotivation, detailreiche Grafik. Mit Level-Editor.

Super Rubik's Cube: Der berühmte "Zauberwürfel" auf dem CPC-Bildschirm. Mit Spiegeln für die verdeckten Würfelseiten. Für Freunde verzwickter Kombination.

Alien Attack: Auch das klassische Ballerspiel für die Arbeitspause darf nicht fehlen. Diese "Space Invaders"-Version überzeugt durch ihre liebevoll gestaltete Grafik.

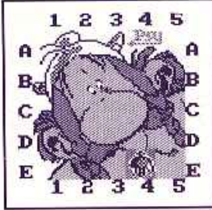
Oik: Temporeiches, "Pacman"-artiges Actionspiel mit niedlich animierter, butterweicher Sprite-Grafik und vielen, vielen Monstern.

Dallas Oil: Management-Gesellschaftsspiel für bis zu 4 Personen. Reizvoll durch ineinander verschachtelte Transaktionen.

Reparator: Flottes Taktik-und Action-Spiel mit farbiger Character-Grafik. Beherrschen Sie das Schlachtfeld?

Gardner-Spiel: Kombination ist angesagt bei diesem Spiel mit Vektoren. Diagonale, senk- oder waagerechte Reihen von Steinen werden auf einem Spielfeld manipuliert. Etwas für Tüftler.

Puzzle — make it!



Neun verschiedene Schiebe-Puzzles zum Entspannen: von den "Ghostbusters" über Conan und Obelix bis hin zum Madonna-Porträt.

255 Schwierigkeits-Stufen. Testen Sie Ihre Vorstellungskraft! **Best.-Nr. 196, DM 15,-***

Codex Vol. 5

Ein Anwenderpaket für kühle Rechner und Freunde des geschriebenen Wortes. **Best.-Nr. 265, DM 19,-***

Textmaker: Ein einfaches Textverarbeitungsprogramm zum Briefeschreiben. Briefkopf gesondert erfaßbar. Blocksatz, Datei-Manager, einstellbarer Tabulator. Zeilenbreite bis 72 Zeichen. Deutsche Schreibmaschinen-Tastaturbelegung mit DIN-Zeichensatz. Basic-Programmierer können die Routinen für deutsche Umlaute und ß auskoppeln und in ihre eigenen Programme einbinden.

Papermaker: Ein Mini-DTP-Programm für die Schüler- oder Vereinszeitung. Mehrspaltiger Text, Schlagzeilen und Zeitungskopf lassen sich frei auf der Seite plazieren. Einfache Geometriegrafik für Zeitungskopf, Köpfe, Schlagzeilen und Artikel getrennt speicherbar. Ausgabe für Epson-kompatible Drucker. Mit einer Demo-Zeitung als Beispiel.

Typograph: Mit diesem Programm entwerfen und verarbeiten Sie Grafikschriften auf dem Bildschirm. Per eingebauter Hardcopyroutine läßt sich die Schriftkunst auch zu Papier bringen. Freies Skalieren und Positionieren der Schriftzeichen. Ein kompletter Font im "Western-Look" wird mitgeliefert — damit Sie gleich loslegen können.

Multitrainer: Ein komfortables Lern-, Übungs- und Test-Programm für beliebige Multiple-Choice-Aufgaben — beispielsweise Führerscheinfragen, Prüfungsaufgaben, Checklisten oder Psycho-Tests. Ergebnisprotokolle speicherbar, Zeitrahmenüberprüfung eingebaut.

THE ESCHWEGE TIMES (1)
frei und ungebunden — Sonderausgabe 14.7.91
HUND MIT ACHT BEINEN — NEUES DTP-PROGRAMM
FREIBIER — FAULPELZE ENTLASTET — HUNDSTAGE

Buchomat: Verwaltung aller privaten Finanzen auf kaufmännischer Basis mit allem Komfort. Komplette Übersicht über Aktiva und Passiva. Kontenrahmen mit insgesamt 28 Konten — auch Wertpapiere und Optionen, Steuern, Gebühren, Abschreibungen und Zinsen werden erfaßt. Jahresabschluß, Druckerausgabe für Gesamt- und Vermögensübersicht.

Haushaltsbilanz: Erfassen Sie Ihre laufenden Ausgaben und stellen Sie fest, in welchen Bereichen die Kostenlawine am schlimmsten wütet. Übersichtliche Tabellendarstellung, blitzschneller Überblick über die finanzielle Situation im laufenden Monat. Kinderleichte Menüsteuerung, Druckerausgabe.

Codex Vol. 6

Eine besonders anspruchsvolle Sammlung hilfreicher Anwendungen für verschiedene Lebenslagen. **Best.-Nr. 266, DM 19,-***

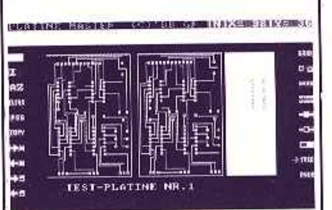
Diskettensystem: Die langgesuchte "eierlegende Wollmilchsau". Von den Standard-Datei-Operationen (Formatieren, Kopieren, Directory-Manipulationen) bis hin zum Diskettenoptimierer, File-Reaktivierer und Relokator ist alles eingebaut. Eine Werkbank für die tägliche Arbeit mit Disketten und Dateien. Ein eigenes Benutzermenü läßt sich einklinken — eine Overlay-Datei mit entsprechender der Schnittstelle ist dabei.

Platine-Master: Ein sehr aufwendiges Programm zur Gestaltung von Leiterplatten. Joystick-Bedienung mit Icons, Fadenkreuzcursor, Masken für IC-Fassungen auf Abruf, Beschriftung, Kopieren von Bereichen, abschaltbares Hilfsraster. Druckausgabe in zwei Größen.

Programmsammlung
Der Datengarten
mit der zusätzlichen Portion
Gebrauchsnutzwert!
3-Zoll-Diskette — CPC international
Das Beste von RxE Software
Die DMV-Super-Serie

Terminkalender: Speichert für jeden Tag einen Textbildschirm. Auch für Tagebuch oder Besuchsprotokolle geeignet. Menügesteuert, mit deutschen Umlauten, Druckerausgabe, zahlreichen Editorfunktionen, Hilfebildschirm.

Magic Synthesizer: Hüllkurven und Tonschwebungen erzeugen für ausgefeilte Sounds mit dem Soundchip des CPC. Grafische Darstellung der Hüllkurven, Schritt-für-Schritt-Editieren durch numerische Eingabe möglich. Erzeugt Basic-Code für die Verwendung der Sounds in eigenen Programmen. Demo-Melodie zum Austesten eingebaut.



Fußball-Verwaltung: Von der Bundesliga bis zur zweiten Kreisklasse — dieses sehr umfangreiche Programm verwaltet alle Fußballergebnisse komfortabel und erleichtert Tippern die Übersicht. Freie Eingabe und Abspeichern von Ligen und Spielplänen. Das Programm überträgt die einzelnen Paarungen in die Ergebnliseingabe. Die Tabellen lassen sich anzeigen und auf dem Drucker ausgeben. Umfangreiche statistische Auswertung.

Lotto-Commander: Erlaubt die Aufstellung und Gewinnkontrolle jeweils eines Service-Scheins der Spiele "6 aus 49" und "Spiel 77" für mehrere Wochen (besonders preisgünstige Tippmethode).

Diskettenhüllen-Drucker: Druckt Inhaltsverzeichnis, Archivnummer und Datum auf Hüllen für 3-Zoll-Disketten. Für Epson-kompatible Drucker.

Aufkleber-Editor: Zur Herstellung professioneller, klebefertiger Etiketten für 3-Zoll-Disketten. Sehr aufwendig aufgemacht: "What you see is what you get". Fett-, Kursiv-, Magerschrift, doppelt hohe, breite, unterstrichene und kopfstehende Schrift. Gleichzeitige Verwendung von 3 Zeichensätzen möglich. Mit eingebautem Zeichensatz-Editor. Für Epson-kompatible Drucker.



DAS ABO

mit den vielen Vorteilen

- Preisvorteil
- Informations-vorsprung
- Zustellkomfort



Information

über

CPC und PCW Joyce

sammeln Sie mit

- Einzelheften
- Sonderheften
- DATABOX



Software

für

CPC und PCW Joyce

erwerben Sie im

Bestellservice

preisgünstig

und

schnell

Abo-Order Zeitschrift

Hiermit bestelle ich **»CPC International«** für mindestens

☐ **12 Ausgaben** ☐ **6 Ausgaben**

Preise: **BRD**	12 Ausgaben	66,– DM
	6 Ausgaben	33,– DM
Auslandspreise: **Europa**	12 Ausgaben	96,– DM
	6 Ausgaben	48,– DM
Außereuropäisches Ausland	12 Ausgaben	120,– DM
	6 Ausgaben	60,– DM

Lieferung erfolgt ab der nächsterreichbaren Ausgabe.

Abo-Order DATABOX

Hiermit bestelle ich die ☐ **CPC-Databox** ☐ **Joyce-Databo**
für mindestens ☐ **12 Ausgaben** ☐ **6 Ausgaben**
als ☐ Cassette ☐ Diskette 3"

Bezugspreise Inland (einschl. Porto/Verpackung)
Cass. 6 Ausgaben 90,– DM, 12 Ausgaben 180,– DM
Disk 3" 6 Ausgaben 150,– DM, 12 Ausgaben 300,– DM

Bezugspreise Ausland (eingekl. Werte außereuropäisch)
Cass. 6 Ausg. 100,– (120,–) DM, 12 Ausg. 200,– (240,–) DM
Disk 3"6 Ausg. 160,– (180,–) DM, 12 Ausg. 320,– (360,–) DM

Lieferung erfolgt ab der nächsterreichbaren Ausgabe.

Lieferanschrift
☐ Geschäft ☐ Privat

Rechner-Typ
☐ CPC ☐ Joyce

Vorname, Name

Firma (nur wenn Lieferadresse)

Straße, Nr./Postfach

PLZ/Ort *(Bitte genaue Anschrift angeben!)*

Dieses Abonnement verlängert sich automatisch um 6 bzw. 12 Ausgaben, wenn es nicht sechs Wochen vor Ablauf schriftlich gekündigt wird.

Datum 1. Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

☐ **Bequem und bargeldlos durch Bankabbuchung**

Bankleitzahl

Konto-Nr./-Inhaber

Geldinstitut

☐ **Gegen Rechnung** – zahlbar innerhalb zwei Wochen nach Erhalt. (Bitte keine Vorauszahlung leisten — Rechnung abwarten.)

Bitte unbedingt auf der Vorderseite eine zweite Unterschrift leisten!

Einzelhefte und DATABOX

Einzelheft Ausgabe 6,– DM/Stck.	CPC Kassette 14,– DM/Stck.	CPC 3"-Diskette 24,– DM/Stck.	Joyce 3"-Diskette 24,– DM/Stck.
1'87 bis 11'88 (Bitte eintragen)			
☐ '	–	☐	☐
☐ 1'89	–	☐	☐
☐ 2'89	–	☐	☐
☐ 3'89	–	☐	☐
☐ 4'89	–	☐	☐
☐ 5'89	–	☐	☐
☐ 6'89	☐	☐	☐
☐ 7'89	☐	☐	☐
☐ 8'89	☐	☐	☐
☐ 9'89	☐	☐	☐
☐ 10'89	☐	☐	☐
☐ 11'89	☐	☐	☐
☐ 12'89	☐	☐	☐
☐ 1'90	☐	☐	☐
☐ 2'90	☐	☐	☐
☐ 3'90	☐	☐	☐
☐ 4'90	☐	☐	☐
☐ 5'90	☐	☐	☐
☐ 6/7'90	–	☐	☐
☐ 8/9'90	–	☐	☐
☐ 10/11'90	☐	☐	☐
☐ 12/1'90/91	☐	☐	☐
☐ 2/3'91	☐	☐	☐
☐ 6/7'91	☐	☐	☐
☐ 12/1'91/92	☐	☐	☐

\+ Porto/Verpackung
(Inland 4,– DM, Ausland 6,– DM) ________ DM

Sonderheft und DATABOX

CPC-Sonderheft DM 14,–	DATABOX 3-Zoll-Diskette	Kombipack Sonderheft + DATABOX 3-Zoll-Diskette DM 29,–
☐ 309 Nr. 3/86	☐ 319 DM 29,–	☐ 3044
☐ 311 Nr. 5/87	☐ 334 DM 38,–	☐ 3045
☐ 312 Nr. 6/88	☐ 340 DM 38,–	☐ 3046
☐ 313 Nr. 7/88	☐ 3135 DM 38,–	☐ 3047
☐ 3136 Nr. 8/89	☐ 3142 DM 38,–	☐ 3048

☐ *Best.-Nr. 3043* **CPC-Sonderheft-Kraftpaket**
5 Stck. CPC-Sonderhefte 3, 5, 6, 7, 8
DM 35,–

☐ *Best.-Nr. 3042* **Sammelpack 1988**
12 Ausgaben PC Amstrad International
1/88 bis 11/88 + 1/89
DM 39,–

Gesamtbetrag: ________ DM

CPC-Bestellservice

3"-Diskette	Kassette		DM
☐ 207	–	**ConText CPC,** Textverarbeitung	59,–
☐ 262		**Der Finanz-Fachmann**	49,–
☐ 202	–	**CopyShop,** Hardcopy-Programm	49,–
☐ 267		**Lernen mit Spaß Nr. 1**	24,95
☐ 268		**Lernen mit Spaß Nr. 2**	24,95
☐ 214	–	**Turbo Data CPC**	69,–
☐ 205	–	**Special Offer 2**	59,–
☐ 30700		**DMV-Überraschungspaket für CPC**	49,–
	☐ 30701	**DMV-Überraschungspaket für CPC**	29,–
☐ 213	–	**Software-Experiment**	39,–
☐ 209	–	**FaBaCom,** Basic-Compiler	49,–
☐ 104	–	**Startest**	19,–
☐ 196		**Puzzle – make it**	15,–
☐ 1011	–	**Fantastic Four,** Spielesammlung	29,–
☐ 107	–	**Special Offer 3,** Spielesammlung	39,–
☐ 1369	–	**Faszination 3D**	39,–
☐ 1012	–	**Game-Box III**	29,–
☐ 106	–	**Know CPC**	29,–
☐ 211	–	**Fraktal Generator 3D CPC**	49,–

\+ Porto/Verpackung
(Inland 4,– DM, Ausland 6,– DM) ________ DM

Joyce-Bestellservice

3"-Diskette	Joyce-Software	DM
☐ 215	**Volume 1,** Charakter-Designer	49,–
☐ 216	**Volume 2,** Dateiverwaltung	49,–
☐ 217	**Volume 3,** GSXplot	59,–
☐ 219	**Volume 4,** Bild-Editor	49,–
☐ 220	**Volume 5,** Datenbank	69,–
☐ 221	**Volume 6,** Tabellenkalkulation	59,–
☐ 222	**Volume 7,** Grafische Benutzeroberfläche	69,–
☐ 194	**The Player's Dream Vol. 3**	19,–
☐ 195	**The Player's Dream Vol. 6**	19,–
☐ 265	**Codex Vol. 5**	19,–
☐ 266	**Codex Vol. 6**	19,–
☐ 262	**CPC-Highlights I**	49,–
☐ 263	**CPC-Highlights II**	49,–
	Anwender-Spiele	
☐ 193	**Joyce Highlights I**	29,–
☐ 264	**Joyce Highlights II**	29,–
	Joyce-Sonderheft-Pakete	
☐ 3050	Sonderheft 2 + DATABOX, 2 Disketten 3"	29,–
☐ 3051	Sonderheft 3 + DATABOX, 2 Disketten 3"	29,–
☐ 3052	Sonderheft 4 + DATABOX, 2 Disketten 3"	29,–
	Joyce-Bücher	
☐ 406	**Joyce – mehr als ein Textsystem + DATABOX,** Diskette 3"	49,–

Gesamtbetrag: ________ DM



Abonnement

Rechnungsanschrift

Vorname, Name

Straße, Nr./Postfach

PLZ/Ort

Widerrufsrecht

Ich weiß, daß ich diese Vereinbarung innerhalb einer Woche beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich widerrufen kann, wobei bereits die rechtzeitige Absendung meines Widerrufschreibens zur Fristwahrung ausreicht. Ich bestätige das durch meine zweite Unterschrift.

Datum 2. Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

Bitte ausreichend frankieren

DMV-Verlag
CPC International
Postfach 250

3440 Eschwege



1. Abonnieren

Alle 2 Monate bietet Ihnen PC International aktuelle Informationen zu CPC und Joyce. Nutzen Sie dieses Wissen mit einem Abonnement zu Ihrem ganz persönlichen

- **Preisvorteil**
- **Informationsvorsprung**
- **Zustellkomfort**

Zur Erinnerung:
Mein persönliches Abonnement wurde bestellt am

Widerrufsrecht

Ich weiß, daß ich diese Vereinbarung innerhalb einer Woche beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich widerrufen kann, wobei bereits die rechtzeitige Absendung des Widerrufschreibens zur Fristwahrung ausreicht.



Einzelhefte

Ja, ich mache von Ihrem Angebot Gebrauch und bestelle die umseitig ausgewählten Produkte. Bitte liefern Sie an folgende Anschrift:

Name

Straße/Postfach

PLZ/ORT

☐ Den Betrag bezahle ich mit dem beigefügten Verrechnungsscheck.
☐ Ich bitte um Lieferung per Nachnahme zuzüglich der Nachnahmegebühr (nur innerhalb der BRD).

Datum Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

Bitte ausreichend frankieren

DMV-Verlag
CPC International
Postfach 250

3440 Eschwege

2. Sammeln

PC-Amstrad-Leser sind immer top-informiert. Daher sollte Ihnen kein Heft oder Sonderheft in Ihrer Sammlung fehlen. Das Eintippen von Listings und Programmen ersparen Sie sich durch unseren günstigen DATABOX-Service. Immer gut beraten sind Sie mit PC Amstrad

- **Sonderheften**
- **DATABOXEN**
- **Sammelpacks**

Zur Erinnerung:
Meine Hefte, Sonderhefte und DATABOXEN wurden bestellt am

Amstrad
INTERNATIONAL
CPC • PCW JOYCE

CPC-/PCW-Joyce-Bestellservice

Ja, ich mache von Ihrem Angebot Gebrauch und bestelle die umseitig ausgewählten Produkte. Bitte liefern Sie an folgende Anschrift:

Name

Straße/Postfach

PLZ/ORT

☐ Den Betrag bezahle ich mit dem beigefügten Verrechnungsscheck.
☐ Ich bitte um Lieferung per Nachnahme zuzüglich der Nachnahmegebühr (nur innerhalb der BRD).

Datum Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

Bitte ausreichend frankieren

DMV-Verlag
CPC International
Postfach 250

3440 Eschwege

3. Bestellen

Die Power-Software von PC Amstrad für CPC und PCW Joyce ist unter Kennern bereits ein Begriff, und unser Sortiment wird Monat für Monat gepflegt und erweitert. Und vergessen Sie nicht: Software von PC Amstrad ist auch ein Geschenk, das ankommt. Nutzen auch Sie die Vorteile von PC-Amstrad-Software

- **preisgünstig**
- **bewährt**
- **praxisgetestet**

Zur Erinnerung:
Meine CPC- und PCW-Joyce-Software wurde bestellt am



Sonderhefte • Sammelordner • Bücher

Ausgabe DOS EXTRA DM 18,–		DATABOX 2 Stk. 5 1/4"-Disk DM 35,–	DATABOX 1 Stk. 3 1/2"-Disk DM 35,–
☐ 356	Nr. 3	☐ 357	☐ 358
☐ 381	Nr. 4	☐ 3811	☐ 3812
☐ 382	Nr. 5	☐ 3813	☐ 3814
☐ 383	Nr. 6	☐ 3815	☐ 3816
☐ 384	Nr. 7	☐ 3817	☐ 3818
☐ 385	Nr. 8	☐ 3819	☐ 3820
☐ 387	Nr. 10	☐ 3823	☐ 3824
☐ 388	Nr. 11	☐ 3825	☐ 3826
☐ 389	Nr. 12	☐ 3827	☐ 3828
☐ 391	Nr. 13	———	———
☐ 392	Nr. 14	☐ 3829	☐ 3830
☐ 393	Nr. 15	☐ 3831	☐ 3832
☐ 394	Nr. 16	☐ 3833	☐ 3834
WinDOS DM 18,–		**1 Stk. 5 1/4"-Disk** DM 24,–	**1 Stk. 3 1/2"-Disk** DM 24,–
☐ 30200	Nr. 1	Diskette im Heft	———
☐ 30201	Nr. 2	Diskette im Heft	———

DMV-ComputerWissen DM 18,–		5 1/4"-Diskette DM 35,–	3 1/2"-Diskette DM 35,–
☐ 3511	Band 1, Basic Toolbox	☐ 3512	☐ 3513
☐ 3514	Band 2, Textverarbeitung	☐ 3515	☐ 3516

	Best.-Nr.		DM
☐	60500	DOS-Taschenbuch Band 1	29,80
☐	60510	DOS-Taschenbuch Band 2	29,80
☐	60520	DOS-Taschenbuch Band 3	29,80
☐	60530	DOS-Taschenbuch Band 4	29,80
☐	40101	Das große DOS-Profi-Arbeitsbuch	78,–
☐	40102	Datenfernverarbeitung – professionell	48,–
☐	40103	Clipper 5.0	58,–
☐	40200	PC Kompakt – ConText 4.0	19,80
☐	40210	PC Kompakt – Windows 3.0	19,80

DOS-Taschenbücher	5 1/4"-Diskette DM 24,–	3 1/2"-Diskette DM 24,–
BAND 1	☐ 60501	☐ 60502
BAND 3	☐ 60521	☐ 60522
BAND 4	☐ 60531	☐ 60532

Sammelordner — *2 Stk. 15,80*

☐ 530 **DOS**
☐ 534 **toolbox**
☐ 529 **WinDOS**
☐ 531 **CPC International**
☐ 535 **PCpur**
☐ 537 **AMIGA DOS**

Zwischensumme + Porto und Verpackung
(Inland 4,– DM, Ausland 6,– DM) ________________

Gesamtbetrag: ________________ DM

PC-Software

vom

DMV Versandservice

- leistungsstark
- praxisbewährt
- preisbewußt

PC-Software

5 1/4"	3 1/2"		DM
2726 ☐	2727 ☐	**ConText 4.0,**	249,–
2728 ☐	2729 ☐	**ConText 4.0-Update**	99,–
2532 ☐	2533 ☐	**ConBase**	149,–
2538 ☐	2539 ☐	**ConView**	49,–
2750 ☐	2751 ☐	**ConLex –** Rechtschreiblexikon Deutsch	49,–
2752 ☐	2753 ☐	**ConLex –** Übersetzungshilfe für Einsteiger Deutsch/Englisch	99,–
2754 ☐	2755 ☐	**ConLex –** Übersetzungshilfe für Einsteiger Englisch/Deutsch	99,–
2756 ☐	2757 ☐	**ConLex –** Übersetzungshilfe für Sprachkenner Deutsch/Englisch	99,–
2758 ☐	2759 ☐	**ConLex –** Übersetzungshilfe für Sprachkenner Englisch/Deutsch	99,–
2760 ☐	2761 ☐	**GRIPS-Desktop und -DeskDraw**	199,–
2534 ☐	2535 ☐	**DMV-ProfiTools** f. Clipper	199,–
2536 ☐	2537 ☐	**DMV-Integrator** f. Clipper	199,–
256 ☐	257 ☐	**PC-VirusDoktor**	99,–
2903 ☐	2904 ☐	**PC-VirusFinder**	149,–
2905 ☐	2906 ☐	**GraphBas PC,** Turbo Basic	99,–
2907 ☐	2908 ☐	**GraphBas PC,** Microsoft Quick Basic	99,–
2522 ☐	2521 ☐	**ERGO 3.0,** DOS-Benutzeroberfläche	199,–
238 ☐	239 ☐	**MaskEdit Plus** mit einem Treiber	99,–
2613 ☐	2614 ☐	**MaskEdit Plus,** Paket mit zwei Treibern	149,–
2621 ☐	2622 ☐	**MaskEdit Plus,** Paket mit drei Treibern	199,–

Turbo Pascal 3/4/5/6	Turbo C Microsoft C	Top-Speed Modula-2	Turbo Basic	Quick Basic
☐ 2632	☐ 2633	☐ 2634	☐ 2635	☐ 2636

5 1/4"	3 1/2"		DM
2720 ☐	2721 ☐	**WinBasic**	299,–
2730 ☐	2731 ☐	**DMV-WindowsTools** SystemInfo	99,–
2732 ☐	2733 ☐	**DMV-WindowsTools** MultiCopy	99,–
2734 ☐	2735 ☐	**DMV-WindowsTools** HexEdit	99,–
2736 ☐	2737 ☐	**DMV-WindowsTools** TextSearch	99,–
2724 ☐	2725 ☐	**ColoRIX**	299,–
6031 ☐	6032 ☐	**ProfEd**	99,–
227 ☐	233 ☐	**BCI DisAsm/I86,** (MS-DOS)	199,–
2738 ☐	2739 ☐	**NeoGraph**	99,–
6017 ☐	6018 ☐	**3D-Draw Professional**	199,–
6019 ☐	6020 ☐	**Navigator 4.5**	69,–
6025 ☐	6026 ☐	**Lightning 3D, 1.5**	199,–

5 1/4"	3 1/2"		DM
6035 ☐	6036 ☐	**Fontline**	299,–
6027 ☐	6028 ☐	**DOS-CAD 3.0**	199,–
2700 ☐	2701 ☐	**DMV-Faktura**	398,–
2710 ☐	2711 ☐	**DMV-Atelier**	249,–
2712 ☐	2713 ☐	**DMV-Atelier** Zusatz-Fonts	99,–
2714 ☐	2715 ☐	**DMV-Atelier** Zusatz-Symbole	99,–
20600 ☐	20601 ☐	**MultiCopy**	29,–
20602 ☐	20603 ☐	**DataBase**	29,–
20604 ☐	20605 ☐	**RAM Patroulile**	29,–
20606 ☐	20607 ☐	**Tippsi**	29,–
20608 ☐	20609 ☐	**Menue**	29,–
20610 ☐	20611 ☐	**Terminplaner**	29,–
20612 ☐	20613 ☐	**Astro**	29,–
20614 ☐	20615 ☐	**TexToCom**	29,–
248 ☐	249 ☐	**Fraktal-Generator 3D,** MS-DOS	69,–
1381 ☐	1382 ☐	**Oil Imperium**	39,95
1383 ☐	1384 ☐	**Loom**	69,95
1385 ☐	1386 ☐	**Zak McKracken**	69,95
1387 ☐	1388 ☐	**Maniac Mansion**	69,95
1295 ☐	1296 ☐	**Memo**	69,–
1297 ☐	1298 ☐	**Trademaster**	49,–
1408 ☐	1409 ☐	**Wing Commander**	89,–
1410 ☐	1411 ☐	**The Secret Missions**	39,–
1406 ☐	1407 ☐	**Wing Commander + The Secret Missions**	119,–
1402 ☐	1403 ☐	**The Secret of Monkey Island**	89,–
1404 ☐	1405 ☐	**Wonderland**	99,–

+ Porto/Verpackung
(Inland 4,– DM, Ausland 6,– DM) ________________ DM

= Gesamtbetrag: ________________ DM



Kleinanzeige

Eine Gelegenheitsanzeige in unserem Kleinanzeigen-Markt macht sich immer bezahlt, ob Sie tauschen, verkaufen oder erwerben wollen!

Und so wird's gemacht:
Kreuzen Sie bitte an, in welcher Rubrik Ihre Anzeige erscheinen soll und ob Sie privat oder gewerblich ist. Dann schreiben Sie Ihren Text so in das vorgezeichnete Feld, daß jeder Buchstabe, jedes Satzzeichen oder jeder Wortzwischenraum ein markiertes Kästchen ausfüllt. Jetzt brauchen Sie nur noch die Zeilen zu zählen, den Preis zu berechnen, einen Verrechnungsscheck auszufüllen, und fertig ist Ihre Gelegenheitsanzeige.

Bitte beachten Sie!
Aus verwaltungstechnischen Gründen kann der Abdruck Ihrer Kleinanzeige nur gegen Vorkasse erfolgen.

Kleinanzeigen-Markt

Private Anzeigen: Nur DM 5,– je angefangene Zeile.
Geschäftliche Empfehlungen: DM 8,– je angefangene Zeile, zzgl. gesetzlicher Mehrwertsteuer.
Bitte veröffentlichen Sie meine Anzeige in der nächsterreichbaren **CPC Amstrad** für
☐ private Zwecke ☐ gewerbliche Zwecke (gewerbliche Anzeigen werden mit **G** gekennzeichnet)
Das ist der Text: *(Bitte deutlich in Druckbuchstaben schreiben!)*

☐ Die Anzeige soll als Chiffre-Anzeige erscheinen (nur bei Privat-Anzeige). Chiffre-Gebühr 10,– DM inkl. MwSt. zzgl. zum Anzeigenpreis.
In dieser Rubrik:

☐ **Biete an**	☐ **Suche**	☐ **Tausch**	☐ **Stellenmarkt/freie Mitarbeit**
☐ Hardware	☐ Hardware		☐ **Geschäftsverbindungen**
☐ Software	☐ Software		☐ **Verschiedenes**

PC-Sonderhefte

Ja, ich mache von Ihrem Angebot Gebrauch und bestelle die umseitig ausgewählten Produkte. Bitte liefern Sie an folgende Anschrift:

Name

Straße/Postfach

PLZ/ORT

☐ Den Betrag bezahle ich mit dem beigefügten Verrechnungsscheck.

☐ Ich bitte um Lieferung per Nachnahme zuzüglich der Nachnahmegebühr (nur innerhalb der BRD).

Datum Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

Bitte ausreichend frankieren

DMV-Verlag
CPC International
Postfach 250

3440 Eschwege

1. Informieren

DOS-Taschenbücher und DOS-EXTRA-Sonderhefte sind eine zuverlässige Informationsquelle für jeden PC-Besitzer. Nutzen auch Sie dieses Wissen, und vertiefen Sie es mit den dazu angebotenen Programmen aus dem DATA-BOX-Service. Den PC beherrschen mit

- **DOS-EXTRA-Sonderheften**
- **DOS-Taschenbüchern**
- **DOS-DATABOX-Service**

Zur Erinnerung:
Meine DOS-Sonderhefte, DOS-Taschenbücher, DOS-DATABOX wurden bestellt am

PC-Software

Ja, ich mache von Ihrem Angebot Gebrauch und bestelle die umseitig ausgewählten Produkte. Bitte liefern Sie an folgende Anschrift:

Name

Straße/Postfach

PLZ/ORT

☐ Den Betrag bezahle ich mit dem beigefügten Verrechnungsscheck.

☐ Ich bitte um Lieferung per Nachnahme zuzüglich der Nachnahmegebühr (nur innerhalb der BRD).

Datum Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

Bitte ausreichend frankieren

DMV-Verlag
CPC International
Postfach 250

3440 Eschwege

2. Aufrüsten

PC-Software vom DMV-Verlag ist unter Kennern bereits ein Begriff und auch ein gutes Angebot für Sie, Ihren PC mit starken und anwendergerechten Applikationen auszustatten. Vom der Textverarbeitung bis zum Programmiersystem, beim DMV-Versandservice kaufen Sie günstig ein. Den PC konsequent nutzen mit PC-Software aus dem DMV-Versandservice

- **PC-Anwendungen**
- **PC-Applikationen**
- **PC-Entwicklungssysteme**

Zur Erinnerung:
Meine PC-Software wurde bestellt am



Kleinanzeige

Bei Angeboten:
Ich bestätige, daß ich alle Rechte an den angebotenen Waren besitze.

Name

Straße/Postfach

PLZ/ORT

☐ Den Betrag bezahle ich mit dem beigefügten Verrechnungsscheck.

Datum Unterschrift (bei Minderjährigen des gesetzlichen Vertreters)

Bitte ausreichend frankieren

DMV-Verlag
CPC International
Postfach 250

3440 Eschwege

3. Chancen nutzen

Wollen Sie etwas aus dem Computerbereich tauschen, verkaufen oder erwerben, dann schalten Sie doch eine Gelegenheitsanzeige in unserem Kleinanzeigen-Markt.

Dabei können Sie bestimmt so manche Mark sparen und ein gutes Schnäppchen machen. Ihre Chancen nutzen Sie konsequent mit

- **Gelegenheitsanzeigen**
- **Kleinanzeigen**
- **Gewerblichen Angeboten**

Zur Erinnerung:
Meine Kleinanzeige wurde aufgegeben am

CPC-Sonderhefte



CPC-Sonderhefte

Randvoll mit nützlichen Tips und Tricks für Ihren CPC-Computer, sollten die Sonderhefte der PC Amstrad International in keiner Sammlung fehlen. Ob komplett als Kraftpaket oder als Kombipack mit der DATABOX - jetzt können Sie kompaktes CPC-Wissen so günstig wie noch nie erwerben.

CPC-Sonderheft-Kraftpaket

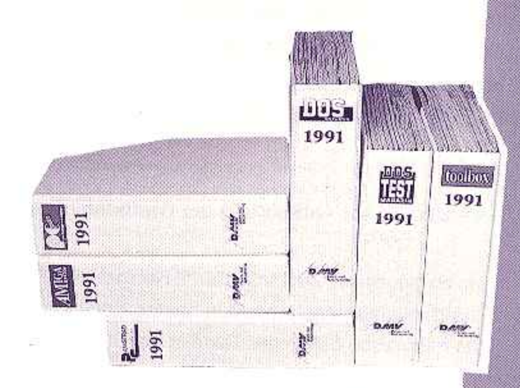
Fünf CPC-Sonderhefte
3, 5, 6, 7, 8 DM **35,-***

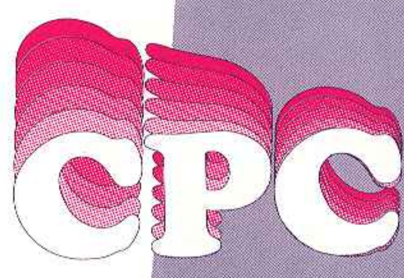
CPC-Sonderheft-Kombipack

Ein Sonderheft (3, 5, 6, 7, oder 8)
+ DATABOX DM **29,-***

DMV-Sammelordner

2 Stück DM **15,80***







Joyce-Programmsammlungen

GCPM - die grafische Benutzeroberfläche für den Joyce

Schnell, bequem und komfortabel

Beneiden Sie nicht länger die PC-Besitzer! Mit GCPM können Sie nun auch für den Joyce eine komfortable grafische Benutzeroberfläche inklusive Mausbedienung, Uhr, Taschenrechner, Schnittstellensteuerung, Passwortabfrage und vielem mehr erwerben.

GCPM bietet Ihnen unter anderem:
● Steuern Sie alle Funktionen Ihres Joyce wahlweise über Maus oder Tastatur. ● Arbeiten Sie mit der Darstellung aller Dateien in Tabellenform ähnlich LogoScript und den Standardfunktionen wie Kopieren, Umbenennen und Löschen. ● Auf bis zu drei Laufwerken können Sie Parameter, Suchpfade (ohne SETDEF) und das temporäre Laufwerk beliebig einstellen. ● Verändern Sie Dateiattribute, und rufen Sie Programme direkt von GCPM aus auf. ● Nutzen Sie die Funktionen zum Verschlüsseln und Komprimieren von ASCII-Dateien und die optionale Passwortabfrage ● Sparen Sie Zeit mit dem integrierten Taschenrechner mit Zwischenspeicher und umfassenden Rechenfunktionen ● Verändern Sie beliebig die Grundeinstellungen der Schnittstelle, des Zeichensatzes, der Tastatur, der Maus, der Diskettenlaufwerke und des Druckers. ● Ein Screensaver stellt den Monitor bei Nichtgebrauch dunkel. ● Mit dem GCPM-Starterset als Installationsprogramm und 40seitigem Handbuch mit vielen Abbildungen können Sie im Handumdrehen GCPM auch auf Ihrem Joyce anwenden. Ein Programm, das jeder Joyce-Besitzer haben muß!

Keine Inhaltsverzeichnisse mehr mühsam einzeln auslesen! Mit GCPM sind Sie über alle Laufwerke immer im Bilde.

Joyce-Programmsammlung 7:
GCPM - Grafische Benutzeroberfläche nur DM **69,-***

Joyce-Bild-Editor

Grafikprogramm zum Erstellen hochauflösender Grafiken

Der Joyce ist mehr als ein Textverarbeitungssystem! Mit dem Joyce-Bild-Editor erstellen Sie komfortabel hochauflösende Grafiken, binden diese in eigene Programme ein oder geben sie auf dem Joyce-Drucker aus.

Der Joyce-Bild-Editor - erschließt die Grafikfähigkeiten Ihres Joyce:

- Exaktes, pixelgenaues Zeichnen durch Normal- und Feinmodus
- Lösch- und Invertierungsfunktion in beiden Bearbeitungsmodi
- Einfache Integration beliebiger Texte in die Bildelemente
- Laden und Speichern von Bildern mit übersichtlichem Dateimenü
- Option zum Mischen von fertigen Bildern nach der Erstellung
- Ausdruckmöglichkeit mit Randbestimmung auf dem Joyce-Drucker
- Bis zu vier verschiedene, vordefinierte Druckformate anwählbar
- Kompletter Quelltext in Turbo Pascal 3.0 wird mitgeliefert

Joyce-Programmsammlung 4:
Bild-Editor nur DM **49,-***

Joyce-Busineß-Paket

Erstellung von Geschäftsgrafiken und Tabellentextverarbeitung

Für Präsentationen erstellen Sie mit GsxPlot Geschäftsgrafiken und Statistiken in Form von Balken-, Kurven- und Punktediagrammen für Bildschirm und Drucker. Mit FeldTab füllen Sie Vordrucke, Formulare und Tabellen mit Ihren Texten aus.

GsxPlot - die Lösung für Busineß-Grafik auf dem Joyce:

- Menügesteuerte Erstellung von Balken- und Kurvendiagrammen
- Strich- und Textgrafik ebenfalls über Menüs steuerbar
- Online-Hilfe zur schnellen Einarbeitung in die Programmfunktionen
- Darstellung der erstellten Grafiken auf Bildschirm und Drucker

FeldTab - plaziert Ihre Texte millimetergenau:

- Komplette Menüsteuerung im gewohnten Aufbau von LocoScript
- Genaue Eingabe von Seitenlänge, Zeilenabstand und Tabulatoren
- Individuelle Bestimmung von Schriftweiten und Schriftarten
- Eingebauter, komfortabler Seiteneditor zur Eingabe der Texte
- Textübernahme von LocoScript oder als ASCII-File möglich

Joyce-Programmsammlung 3:
Busineß-Paket nur DM **59,-***

Hochwertige Software zu Niedrigpreisen für Joyce-Besitzer aus der Angebotspalette des DMV-Verlags!

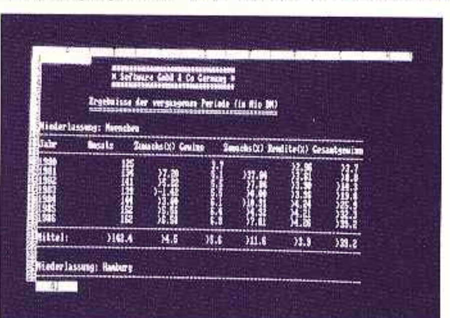
Jede Programmsammlung ist, soweit nicht anders vermerkt, auf Joyce PCW 8256 und 8512 lauffähig und wird als auf 3-Zoll-Diskette mit deutscher Bedienungsanleitung ausgeliefert.

Joyce-Tabellenkalkulation

Die universelle Tabellenkalkulation für alle Joyce-Besitzer. Ob Monatsbilanzen, Steuerabrechnungen oder Kontoführung – mit der Joyce-Tabellenkalkulation sind Sie allen Aufgaben gewachsen.

Schnell, einfach und sicher - die Joyce-Tabellenkalkulation:

- Schnelle Dateioperationen durch Benutzung der RAM-Disk
- Besonders schnelle Berechnungsalgorithmen für die Felderwerte
- Übersichtliche Menüführung und Abfangen von Fehleingaben
- Standardmäßig bis zu 2574 frei belegbare Felder für Berechnungen
- Am Rand eingeblendete Formel mit bis zu 68 möglichen Zeichen
- Eigene Befehle zur Verkürzung der Formeln und Exponentialausdrücke
- Ausdruck der Ergebnisse in bis zu drei verschiedenen Schriftarten



Joyce-Programmsammlung 6:
Tabellenkalkulation für alle Joyce PCWs
nur DM **59,-**

Joyce-Datenbank

Die ideale Datenbank für den Joyce mit minimalem Disketten-Speicherbedarf. Dennoch bietet Ihnen die Joyce-Datenbank maximal 27 Felder pro Datensatz, 50 Stellen pro Feld und bis zu 35.000 Datensätze pro Datei.

Daten effektiv verwalten mit der Joyce-Datenbank:

- Besonders schnell durch Verwendung der RAM-Disk für das Programm
- Universell durch die Verwendung von Standard-Direktzugriffsdateien
- Bequem durch Datentransfer von vorhandenen in neuangelegte Dateien
- Maske zur Datenerfassung und Änderung wird automatisch generiert
- Komfortabel durch Wahl der Sortierkriterien oder Mehrfachsortierung
- Listen mit automatischer Spaltenformatierung und Spaltensummen
- Automatischer Eintrag neuer Dateien in das Disketten-Hauptmenü
- Speicherung einmal gewählter Druckparameter zur Wiederverwendung

Joyce-Programmsammlung 5:
Datenbank (PCW 9512 auf Anfrage)
nur DM **69,-***

Joyce-Zeichensatz-Designer

Erstellung von Zeichensätzen und grafische Darstellung mathematischer Funktionen

Sonderzeichen und ganze eigene Zeichensätze für die Bildschirmausgabe erstellen Sie komfortabel und einfach mit dem Joyce-Zeichensatz-Designer. Mit dem Programm MGX können Sie mathematische Funktionen und Meßreihen auf Bildschirm und Drucker ausgeben.

Zeichensätze selbstgemacht mit dem Joyce-Zeichensatz-Designer:

- Editieren, Erstellen und Abspeichern von Bildschirm-Zeichensätzen
- Drucken von vorformatierten ASCII-Texten unter CP/M Plus möglich
- Vier fertig gestaltete Zeichensätze im Lieferumfang enthalten
- Bonusprogramm zum Einstellen der Joyce-Schnittstelle und Tastatur

MGX - Funktionen und Meßwerte grafisch auf Bildschirm und Drucker:

- Arithmetische Grund- und Sonderfunktionen sind darstellbar
- Gleichzeitige Darstellung mehrerer Funktionen und Meßreihen
- Druckerausgabe der Ergebnisse im Großformat möglich
- Neben einer Klammerebene ist eine Memory-Funktion integriert

Joyce-Programmsammlung 1:
Zeichensatz-Designer
nur DM **49,-***

Joyce-Dateiverwaltung

Universelle Erstellung und Verwaltung eigener Dateien

Legen Sie Ihre Daten in eigene Dateien ab, und verwalten Sie diese mit SuperDat. Rundschreiben, Serienbriefe und eine Übernahme der Daten in LocoScript erledigen Sie mit SuperTex. Ein leistungsfähiger Taschenrechner steht Ihnen mit SuperCal zur Verfügung.

SuperDat - Super-Dateiverwaltung für Ihre Daten:

- Komfortable Eingabe der Daten in eine Eingabemaske
- Mehrere Arten von Suchfunktionen, z.B. auch Suche mit Jokern
- Dateien bis zu 8 Feldern und Datensätze bis 255 Zeichen möglich

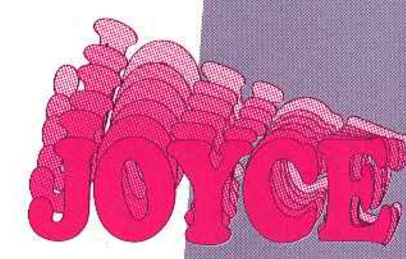
SuperTex - verarbeitet Ihre Daten automatisch:

- Einfügen von ausgewählten Einträgen aus SuperDat-Dateien in Text
- Umwandlung von bis zu 30 Datensätzen in eine LocoScript-Datei

SuperCal - der Tachenrechner für den Joyce:

- Bietet Grundrechenarten, Winkel-, Quadrat- und Prozentfunktionen
- Neben einer Klammerebene ist eine Memory-Funktion integriert

Joyce-Programmsammlung 2:
Dateiverwaltung
nur DM **49,-***





Joyce-Sonderhefte

Als Sonderpublikationen der PC Amstrad International bieten die Joyce-Sonderhefte auf jeweils 120 Seiten ausschließlich erstmalig veröffentlichte Beiträge sowie reichlich Tips & Tricks zu Joyce/PCW 8256, 8512 und 9512.

Aus dem Inhalt:

Joyce-Sonderheft 4:

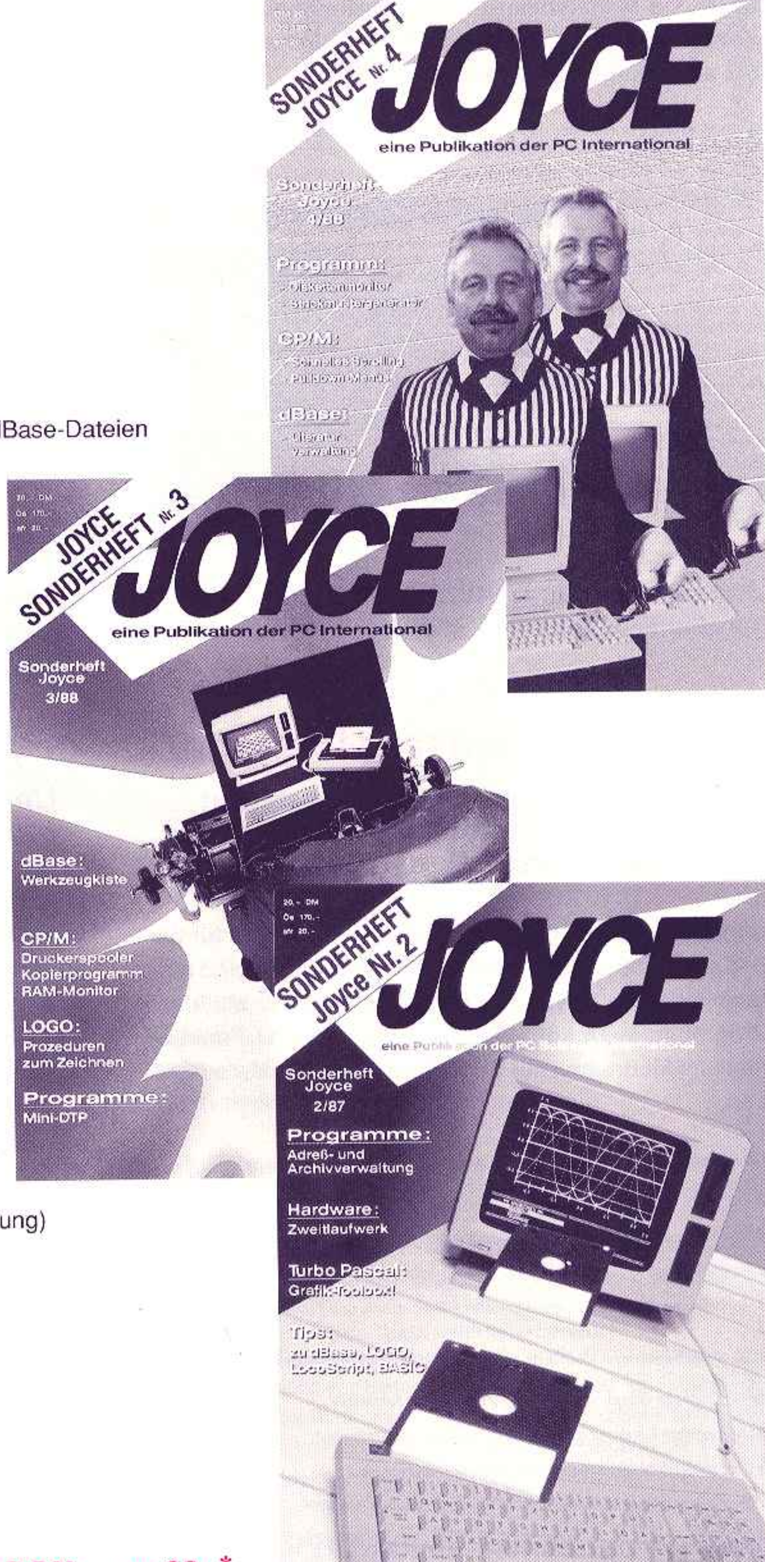
- Strickmustergenerator
- WordStar-Verbesserungen
- Bundesligasimulation
- Super-Reaktionsspiel Filemanager
- Pulldown-Menüs
- Astrologieprogramm
- Diskettenmonitor
- Hauptstädte raten in LOGO
- Statuszeile für dBase und Basic
- Hardcopy-Routine für 24-Nadler
- LOGO macht Schachteln
- dBase-Literaturverwaltung
- Universelles Werkzeug zur Veränderung von dBase-Dateien
- Joyce-Sonderheft-Kombipack

Joyce-Sonderheft 3:

- Vokabeltrainer
- RAM-Monitor – Speicherinhalte verändern
- Memory-Spiel
- Mini-DTP-Programm
- Drucker-Spooler unter CP/M
- Disketten-Kopierprogramm bis 43 Spuren
- Grafik auf dem Joyce-Drucker
- Tastaturbelegung unter CP/M und LocoScript ändern
- Reset ohne Datenverlust
- Super-Werkzeugkiste dBase
- Grafikutilities für LOGO

Joyce-Sonderheft 2:

- Adreßverwaltung
- Archivprogramm (Video- oder Literaturverwaltung)
- Pascal-Compiler in Basic
- Suburbia (Spiel ähnlich Monopoly(R))
- Turbo-Pascal-Grafikroutinen ohne GSX
- Komfortable Balkengrafik
- Joyce-Zweitlaufwerk selbst anschließen
- 3D-Plotter
- Etikettendruckprogramm
- Ordnung auf der Diskette mit LocoScript
- dBase-Handbuch selbst ausdrucken
- LOGO-Funktionenzeichner



Jeweils ein Sonderheft + die DATABOX DM **29,-***

DMV-Bücher zum Joyce

Alle Besitzer eines Joyce-PCW, die ahnten, daß der Horizont Ihres Computers weit über LocoSript hinausgeht, finden jetzt Ihre Bestätigung:

Joyce - mehr als ein Textsystem

Auf über 300 Seiten tragen die Autoren alles Wissenswerte über den Joyce/PCW zusammen. Anfänger wie Profis, Anwender wie Programmierer finden in diesem Buch einen reichen Schatz an Tips und Tricks sowie ein unentbehrliches Nachschlagewerk.

Erstmals wird auch in einem Buch ein »heißes« Eisen ausführlich behandelt: die Hardware des Joyce. Besonders dieses Kapitel zeigt völlig neue Möglichkeiten des Joyce – so wird beispielsweise der Anschluß eines Sprachsynthesizers und der Selbstbau einer Schnittstelle besprochen.

Ausführliche Kapitel über Basic und LOGO erlauben Ihnen, die Möglichkeiten dieser Sprachen auszureizen. Anfänger, die sich erstmals in der Programmierung versuchen wollen, finden durch sinnvolle, kurze Beispielprogramme wertvolle Unterstützung. Ein besonderes »Schmankerl« ist das große Extra-Kapitel zur JETSAM-Dateiverwaltung.



Aus dem Inhalt:

- **Sprachen:**
  LOGO als Grafiker und mit kompletter Befehlsübersicht
  Erläuterung aller Basic-Befehle mit Beispielprogrammen
  Generator für Jetsam-Verarbeitung
  Kurzübersicht für Turbo Pascal und C
- **Programmierhilfen, Tips und Tricks:**
  Steuercode-Tabellen für Bildschirm und Drucker
  XBIOS-Routinen
  OUTs und POKEs unter Basic
- **Hardware:**
  Speichererweiterung
  Zweitlaufwerk
  Druckkopfreinigung
  Bildschirminverter
  Schnittstelle am Expansions-Port
  Sprachsynthesizer

Joyce - mehr als ein Textsystem
325 Seiten mit farbigen Abbildungen, Leinen-Hardcover,
inklusive 3-Zoll-Diskette jetzt nur DM **49,-***

Unser Joyce-Hit weiter im Angebot:

Praktische Textverarbeitung mit Joyce

Der Autor Jürgen Siebert zeigt in diesem Buch die Möglichkeiten der Textverarbeitung LocoScript auf. Darunter vieles, was Sie von LocoScript nicht erwartet hätten....

Von der Pike auf werden Sie an den Umgang mit Schablonen und Standard-Layouts herangeführt. Einige Abstecher führen auch zu anderen Textverarbeitungen unter CP/M wie ED und Wordstar.

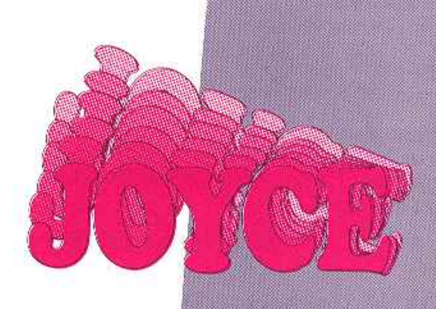
Auf der Diskette erhalten Sie über 50 Dateien mit Schablonen, Brief- und Postkarten-Layouts, Serien-Rundschreiben, Etiketten, Formularen, Schriften, Bildschirminstallationen und vieles mehr.



Aus dem Inhalt:

- LocoScript-Training für Fortgeschrittene
- Wie rette ich den Text bei Systemfehlern?
- Joyce-Tasteninstallation für Wordstar
- Aleatorische Poetik: Der Computer dichtet

Praktische Textverarbeitung mit Joyce
207 Seiten,
Leinen-Hardcover,
Inklusive 3-Zoll - Diskette
jetzt nur DM **49,-***



DATABOX

Für alle CPCs als Kassette und 3-Zoll-Diskette.

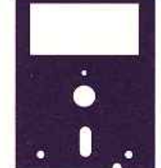
CPC INHALT	DATABOX-Werbung HEFT 12/1'91/92 464	664	6128
RSX-Professional	○	○	●
Des Elektronikers Freund	●	●	●
Alles in Ordnung?	○	○	●
Inline für Z80	○	○	●
Alkatrax-Codiersystem	●	●	●
Gigacad Grafikkonvertierung	○	●	●
1 kByte: Kreisschrift	●	●	●
1 kByte: Zahlenraten	●	●	●
1 kByte: Magic Rings	●	●	●
1 kByte: Schiffe versenken	●	●	●
1 kByte: Mini-Vokabelpauker	●	●	●
1 kByte: Bildschirmwechsel	●	●	●
Checksummer	D A T A B O X		
Emulator	D A T A B O X		
Bonusprogramme			
Vier kByte ASCII-Text-Editor	●	●	●
CopyWrite			

Die DATABOX enthält außerdem noch Info-Files (*.INF).

Das ist die Software zur PC INTERNATIONAL Jeden Monat neu

DATABOX:

- mehr als der übliche Software-Service
- bringt ergänzend sämtliche Listings der jeweiligen Zeitschrift und alle Programmbeispiele auf Kassette oder auf 3-Zoll-Diskette
- Die Programme sind, soweit systembedingt möglich, auf allen drei CPC-Modellen lauffähig. Einzelheiten entnehmen Sie bitte der nebenstehenden Aufstellung.
- Soweit die Programme nicht Bestandteil einer Serie sind, befinden sich alle Programme als »ready to run« auf der DATABOX.
- erscheint jeden Monat und trägt das Titelbild des gleichzeitig erscheinenden Heftes
- Der Datenträger zur PC International enthält außerdem jedesmal ein zusätzliches Bonusprogramm, das nicht im Heft abgedruckt ist.





Einzelbezugspreise für DATABOX:

3-Zoll-Diskette

Inland:		**Ausland:**	
Einzelpreis	24,– DM	Einzelpreis	24,– DM
zzgl. Versandkosten	4,– DM	zzgl. Versandkosten	6,– DM
Endpreis	**28,– DM**	**Endpreis**	**30,– DM**

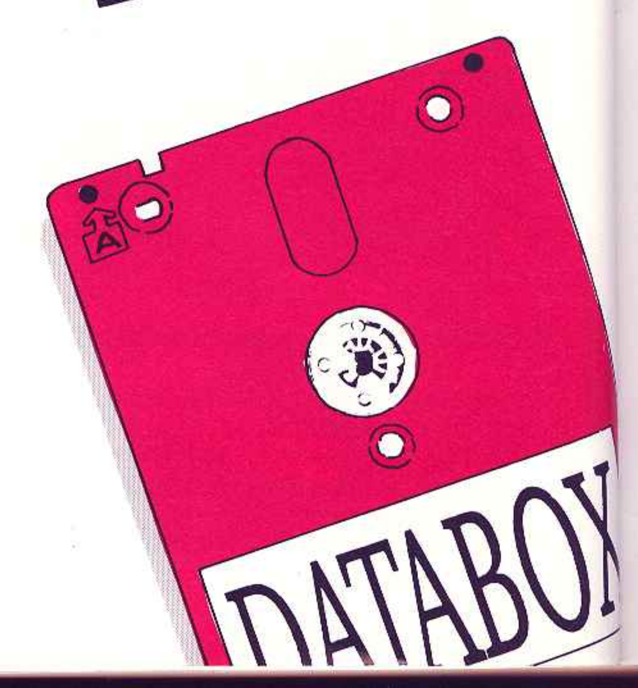
Kassette

Inland:		**Ausland:**	
Einzelpreis	14,– DM	Einzelpreis	14,– DM
zzgl. Versandkosten	4,– DM	zzgl. Versandkosten	6,– DM
Endpreis	**18,– DM**	**Endpreis**	**20,– DM**

Zahlungsweise:
Am einfachsten per Vorkasse (Verrechnungsscheck) oder als Nachnahme zuzüglich der Nachnahmegebühr. Bei Lieferungen ins Ausland ist Nachnahme nicht möglich.

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege



was man wissen muß, aber daß man sich nicht fortbewegen kann, mutet schon merkwürdig an. Tatsächlich kommt man nur nach oben, indem man einen Roboter auf einem Feld erschafft, das etwa auf Augenhöhe des aktuellen Roboters ist, und dann überwechselt. Der verlassene Roboter sollte dann natürlich so schnell wie möglich absorbiert werden. Nur so kann man sich ganz nach oben arbeiten.

Sentinel

Wenn man das Fadenkreuz mit der Leertaste abschaltet, kann sich der Roboter schneller drehen. Vorsicht, wenn man vom *Sentinel* gesehen wird. Wer in Panik kommt und'Hyperraum fährt, hat oft keine Chance mehr. Also schnell handeln und sich einen sicheren Platz aussuchen.

Wer so weit gekommen ist, daß er den *Sentinel* absorbieren kann, sollte seine Euphorie beherrschen und zuvor die Landschaft von oben betrachten. Dies sieht nicht nur schön aus, sondern eröffnet auch den Blick auf Bäume, die noch absorbiert werden können, bevor man den Level verläßt.

Ein Spiel mit Ausblick

In den höheren Levels bekommt der *Sentinel* Unterstützung von Helfern, und man hat kaum Zeit, auf einem Platz zu bleiben. Hier sollte man zuerst Höhe gewinnen und nur so viel Energie wie nötig sammeln. Wenn man erst einmal so hoch ist, daß die Helfer absorbiert werden können, sollte man dies tun. Danach hat man viel mehr Ruhe, die Energie wieder aufzufrischen.

Auch wenn **Rainbow Islands** so süß aussieht, daß man es für ein Spiel für kleine Kinder halten könnte, fordert es doch sogar abgebrühte Spieler heraus. Deshalb sollte man sich auch nicht mit der Jagd nach Bonuspunkten aufhalten, das Zeitlimit ist eng. Anders sieht's aus, wenn es Extras zu erhaschen gibt. Wenn man hier schnell genug ist, erleichtert dies das Spiel ungemein.

Rainbow Islands

Manche meinen, es wäre einfacher, möglichst schnell hochzuklettern und an den Feinden vorbeizuspringen. Man wird aber, ist man erst nach unten gefallen und sozusagen im Wespennetz gelandet, schnell feststellen, daß dies der falsche Weg ist.

Diamanten kommen übrigens besonders dann zum Vorschein, wenn ein Gegner von einem schießenden Stern getroffen wurde. Bei den Gegnern am Ende eines Levels sollte man sich Zeit lassen und studieren, nach welchem Prinzip sie sich bewegen. Die Spinne am Ende des ersten Levels beispielsweise hüpft umher. Und zwar so hoch, daß man darunter hindurch rennen kann – oder vielmehr muß. So gibt es zu jedem Endmonster eine spezielle Taktik. Wenn man diese herausgefunden hat, ist es nur noch halb so schwer.

Will man nach rechts weiterfliegen, steht man bei **Extreme** vor einer schier unlösbaren Situation. Es gibt zwar drei Öffnungen, aber kurz bevor man eine erreicht, wird diese von einer Barriere verschlossen, und man fühlt sich wie Sysiphus. Abhilfe schafft das Turbo-Extra. Man bekommt es, wenn man lange genug auf das windschnittige Raumschiff an der äußersten Ecke (links oben) des Spielfelds schießt. Mit diesem Turbo erreicht man einen Durchgang, ehe er sich verschließt.

Extreme

Zuvor sollte man aber die Energie-Sphären-Kugel holen und abschleppen. Dazu muß man sich durch die Lava kämpfen und dabei in der wandernden, eckigen Luftblase bleiben.

Bei **Pick'n Pile** sollte man sich gar nicht erst bemühen, die Kugeln hoch zu stapeln oder vorauszuplanen. Wer schnell ist, braucht sich nicht darum zu scheren, ob er rationell arbeitet. Lediglich auf die Köpfe sollte man ein Auge haben. Wenn diese den Boden berühren, wird's brenzlig. Auch die Bonus-Zeit einzusammeln lohnt sich. Wenn am Ende nur noch verschiedenfarbige Kugeln übrig sind, einfach so oft ENTER drücken, bis sich alle entfernen lassen.

Strategien und Taktiken gibt es viele bei **Firezone**. Mehr, als sich hier aufzählen lassen. Anfängern als Tip: Man sollte sich in jeder Kampfrunde auf wenige Gegner beschränken, diese dann aber dafür so in die Zange nehmen, daß man deutlich überlegen ist. Besonders wichtig ist, mit den Reichweiten der einzelnen Waffen richtig umzugehen.

Im ersten Raum von **Rick Dangerous II** muß man einfach nur durchlaufen ohne anzuhalten. Dasselbe gilt für die Treppe im zweiten Raum. Wenn man auf Roboter trifft, wird man Schüsse und Bomben schätzen lernen. Diese sollten aber gezielt eingesetzt werden, denn sie sind nicht gerade im Überfluß vorhanden.

Neue Munition gibt es in versteckten Bonus-Räumen. Diese haben oft noch einen zweiten Vorteil: Wenn man ein Leben verliert, darf man dort wieder anfangen. Viele Laser lassen sich durch einen Knopf ausschalten, der irgendwo – vielleicht sogar in einem anderen Raum – versteckt ist. Ansonsten hilft es weiter, darunter durchzukriechen.

Laser einfach ausschalten

Ganz wichtig ist, nicht überstürzt zu handeln. Da es kein Zeitlimit gibt, sollte man lieber etwas langsamer vorgehen, als durch Unachtsamkeit ein Leben zu verlieren. Auch bei diesem Spiel erweist sich das Zeichnen einer Karte, auf der alle versteckten Gemeinheiten eingezeichnet sind, als nützlich. Notfalls reichen auch ein paar Notizen, damit man nicht immer wieder in die gleichen Fallen tappt.

Andreas Lober, jg

Schwierigkeiten?

An dieser Stelle wollen wir uns in Zukunft, im Rahmen der leicht geänderten "Gamers Message", Ihren Problemen mit beliebten Spielen widmen. Schreiben Sie uns doch einmal, bei welchem Spiel Sie an welcher Stelle nicht mehr weiterkommen. Wir bemühen uns, Ihnen zu helfen. Ihre Tips und Tricks zu Spielen sind uns ebenfalls immer willkommen.

Die Adresse:
DMV-Verlag
Red. CPC International
Kennwort: Gamers Message
Fuldaer Straße 6, 3440 Eschwege



GigaCAD

Grafikkonvertierung unter CP/M

Das Programm "ARTWORX" bietet ja bekanntlich die Möglichkeit, GIGA-CAD Bilder zu konvertieren und dann unter "ARTWORX" weiter zu bearbeiten. Nun vermißt man jedoch auch die Möglichkeit, GIGA-CAD Bilder mit anderen Grafikprogrammen weiter zu bearbeiten, oder diese in eigene Programme als Titelscreens einzubinden.

Versucht man ein GIGA-CAD Bild mit der Befehlssequenz: *LOAD* <name.pia>,&C000 in den Bildschirmspeicher zu laden, erhält man die Fehlermeldung "SYNTAX ERROR". Der Grund hierfür liegt auf der Hand: Da GIGA-CAD unter CP/M abläuft, besitzt das erzeugte Bild keinen Datei-Header, in dem wichtige Informationen über die Datei, wie zum Beispiel der File-Typ, die Ladeadresse, und die File-Länge enthalten sind. Um ein GIGA-CAD Bild so zu konvertieren, daß man es unter Basic laden kann, muß es lediglich mit einem Header versehen werden.

Um dies zu bewerkstelligen, muß man folgendermaßen vorgehen:
Man lädt unter CP/M mit einem Debugger ein GigaCAD-Bild an die Adresse 180H ein und füllt dann den Bereich von 100H bis 180H mit Nullstellen auf, um eventuell in diesem Bereich vorhandenen "Müll" zu entfernen.

Daraufhin müssen, wie in den Beispiellistings zu ersehen ist, in die entsprechenden Speicherzellen die Informationen über Filename, Ladeadresse, Filelänge, und Filetyp eingepatcht werden.

Nun kann der Bereich zwischen 100H bis zum Dateiende abgespeichert werden.

Da diese Methode ziemlich umständlich ist, wurden drei SUBmit-Dateien erstellt, die Ihnen die Konvertierung bei den unterschiedlichen CPC Konfiguraltionen wesentlich vereinfachen.

● **GIGAZMON.SUB** (arbeitet unter CP/M PLUS mit dem Public Domain Debugger "ZMON+" von Martin Kotulla)
Bei Verwendung dieser Submit-Datei müssen sich folgende Dateien auf der Diskette befinden: GIGAZMON.SUB, ZMON+.COM, ein GIGA-CAD Bild zum Beispiel DEMO.PIA, und SUBMIT.COM .
Aufruf unter CP/M PLUS:

```
SUBMIT GIGAZMON Bildname Extension
```

● **GIGASID.SUB** (arbeitet unter CP/M PLUS mit dem Debugger "SID", der ja als Dienstprogramm mit CP/M PLUS geliefert wird.
Bei Verwendung dieser Submit-Datei müssen sich folgende Dateien auf der Diskette befinden: GIGASID.SUB, SID.COM, ein GIGA-CAD Bild z.B DEMO.PIA, und SUBMIT.COM .
Aufruf unter CP/M PLUS:

```
SUBMIT GIGASID Bildname Extension
```

● **GIGADDT.SUB** (arbeitet unter CP/M 2.2 mit dem Debugger "DDT")
Bei Verwendung dieser Submit-Datei müssen sich folgende Dateien auf der Diskette befinden: GIGADDT.SUB, DDT.COM, ein GIGA-CAD Bild z.B DEMO.PIA, SUBMIT.COM, und XSUB.COM .
Aufruf unter CP/M 2.2:

```
SUBMIT GIGADDT Bildname Extension
```

Bei allen drei Aufrufen müssen Sie darauf achten, daß der Bildname und die Extension nicht – wie gewohnt – durch einen Punkt, sondern durch ein Leerzeichen getrennt sind.

Den Ausdruck in Mode 2

Ein schönes Ergebnis erhält man, wenn man das konvertierte GIGA-CAD Bild mit dem in der CPC 10/88 veröffentlichten Programm 'Mikro-Copy' ausdruckt, man sollte aber vor dem Laden des Bildes 'MODE 2' wählen, weil dies auch der Mode ist, unter dem das GIGA-CAD Bild erzeugt wird.

Christian Heinz/rs

```
;Programm: GIGAZMON.SUB
;Version für CP/M Plus mit PD Debugger ZMON+
;Aufruf: A>SUBMIT GIGAZMON Bildname Extension
zmon+
<i $1.$2,0180
<y 0100 017F 00
<y 0101 0107 43,4F,4E,56,45,52,54
<y 0108 0108 20
<y 0109 010B 42,49,4E
<y 0112 0112 02
<y 0116 0116 C0
<y 0119 0119 40
<y 0141 0141 40
<y 0143 0144 5C,04
<w $1.BIN 0100 40FF
<^C
```

```
;Programm: GIGASID.SUB
;Version für CP/M Plus mit Debugger SID.COM
;Aufruf: A>SUBMIT GIGASID Bildname Extension
sid
<r$1.$2,0080
<f0100,017F,00
<f0101,0101,43
<f0102,0102,4F
<f0103,0103,4E
<f0104,0104,56
<f0105,0105,45
<f0106,0106,52
<f0107,0107,54
<f0108,0108,20
<f0109,0109,42
<f010A,010A,49
<f010B,010B,4E
<f0112,0112,02
<f0116,0116,C0
<f0119,0119,40
<f0141,0141,40
<f0143,0143,5C
<f0144,0144,04
<w $1.BIN,0100,40FF
<^C
```

```
;Programm: GIGADDT.SUB
;Version für CP/M 2.2 mit Debugger DDT.COM
;Aufruf: A>SUBMIT GIGADDT Bildname Extension
xsub
ddt
i$1.$2
r0080
f0100,017F,00
f0101,0101,43
f0102,0102,4F
f0103,0103,4E
f0104,0104,56
f0105,0105,45
f0106,0106,52
f0107,0107,54
f0108,0108,20
f0109,0109,42
f010A,010A,49
f010B,010B,4E
f0112,0112,02
f0116,0116,C0
f0119,0119,40
f0141,0141,40
f0143,0143,5C
f0144,0144,04
g0
save 64 $1.BIN
```

Bücher unter der Lupe

Schneider CPC Grafik – Einführung, Beispiele, Anwendungen

Autor: J. Hegner
Verlag: IWT Verlag GmbH München
ISBN: 3-88322-147-3

Eine der interessantesten Eigenschaften des CPC dürfte wohl seine Fähigkeit zur Grafikdarstellung sein. Das hier vorgestellte Buch beschäftigt sich mit der Grafikprogrammierung vom BASIC aus. Auf fast 300 Seiten bekommt der Leser eine komplette Einführung in die Handhabung der Grafik-Routinen.

Begonnen wird ganz einfach, nämlich mit dem MODE-Befehl. Anschließend geht Hegner ausführlich auf das Anfertigen von Zeichen- oder Blockgrafiken unter Nutzung des PRINT-Befehls ein. Man lernt den Umfang des bereits im CPC integrierten Zeichensatzes kennen. Wichtige Befehle wie BORDER, INK, PEN und CLG werden erklärt. Die Definition eigener Zeichen mit Hilfe des SYMBOL-Befehls beschließt diesen Abschnitt.

Im nächsten Teil des Buches geht es schon in die Vollen. Hier ist das Thema die hochauflösende Grafik, bei der jeder Punkt des Bildschirms einzeln angesprochen wird. Nach der Klärung wichtiger Begriffe wie etwa "Koordinatensystem" geht es gleich an den ORIGIN-Befehl.

Nun erfährt der Leser einiges über Befehle, mit denen die Manipulation jedes einzelnen Bildpunktes gewährleistet wird. Das alles läßt sich an einfachen, leicht nachvollziehbaren Beispielen üben.

Nun ist etwas Ahnung von Mathematik gefragt – wobei allerdings halbwegs gepflegte Schulkenntnisse völlig ausreichen. Es geht um die Trigonometrischen Funktionen. Mit Hilfe der Funktionen Sinus und Kosinus werden die herrlichsten Grafiken auf den Bildschirm gezaubert. Auch so geheimnisvolle Dinge wie Lissajous-Figuren und Zykloiden werden anhand guter Beispiele erklärt. Ein kleiner Auszug, der zeigt, wie schnell hier das Gelesene am Bildschirm umzusetzen ist:

```
110 MODE 1
120 BORDER 0
130 INK 0,0: INK 1,24
140 DEG
150 FOR j=-40 to 40 STEP 20
160 FOR i= 0 to 359 STEP 0.5
170 x=320+300*sin(i)
180 y=200+180*sin(4*i+j)
190 PLOT x,y
200 NEXT
210 NEXT
220 IF INKEY$="" THEN 220
```

Dieses kurze Programm erzeugt verschiedene Lissajous-Figuren, bei denen sich die Phase von -40^0 in 20^0-Schritten auf $+40^0$ bewegt.

Hegner verwendet dankenswerterweise auch etliche Seiten darauf, die Fenstertechnik zu erklären. Sie stellt eine äußerst effektive Methode dar, Übersichtlichkeit auf den Bildschirm zu bringen.

Obgleich das CPC-BASIC die Fenstertechnik direkt unterstützt, greifen viele BASIC-Freunde aus Unkenntnis nie zu diesem sinnreichen Werkzeug.

Am Schluß des Buches findet sich noch ein Anhang, in dem die wichtigsten Details nachgeschlagen werden können. Neben einem Verzeichnis der Grafik-BASIC-Befehle sind hier noch viele andere wichtige Dinge wie eine Farbtabelle und ein Verzeichnis der Beispielprogramme abgedruckt. Ein solcher Anhang macht Hegners Grafikbuch letzten Endes erst so richtig tauglich für die Praxis.

jg

Z80-Maschinensprachekurs für den CPC 464/664/6128

Autor: Peter Heiß
Verlag: Heise-Verlag, Hannover
Preis: DM 29,80

Dieser Komplett-Kurs für CPC 464/664 und 6128 stellt zwar nicht das erste Assembler-Buch, aber sicher eines der besten dar.

Er wendet sich an alle Programmierer, die bereits erste Erfahrungen mit Basic gesammelt haben und nun tiefer in Rechner und Betriebssystem einsteigen wollen.

Begriffe wie Adressen, Speicher, CPU und Variablen werden als Grundwissen vorausgesetzt. Der Leser wird gleich ins kalte Wasser von Flags, Registern und Byteverknüpfung geworfen, was dem erfahrenen Basic-Programmierer schnell zu sichtbaren Lernerfolgen verhelfen dürfte.

Dieser Kurs ist so ausgelegt, daß Neugierige bald ihre ersten RSX-Befehle implementieren können, ohne sich lange mit Grundlagen und Rechenoperationen aufhalten zu müssen.

Genauso lassen sich aber auch alle Kapitel der Reihe nach abarbeiten, was dann neben dem Erlernen der eigentlichen Assemblersprache auch zu fundiertem Wissen über die Architektur des CPC-Systems führt. Ein Anhang mit kompletter Befehlsliste des Z80-Prozessors macht das Buch auch zu einem wertvollen Nachschlagewerk für die Programmierung.

Die einzelnen Kapitel sind jeweils mit nützlichen kleinen Basic- und Assembler-Programmen gespickt, was schnelle Erfolgserlebnisse garantiert.

Ulrich Schmitz

Kampf dem "GOTO"

Strukturierte Programmierung in BASIC

Programmieren Sie in BASIC? Die meisten CPC-Benutzer haben dies bestimmt schon getan oder haben es zumindestens versucht. Ein paar unentbehrliche Techniken des guten Programmierens wollen wir Ihnen mit diesem Beitrag vermitteln.

Wer kennt ihn nicht , den allseits verwendeten, nicht immer beliebten und doch unentbehrlichen BASIC-Befehl "GOTO"? Dieser Befehl ist hauptsächlich für die zum Teil recht verworrenen BASIC-Programme verantwortlich, zumal Standard-BASIC fast keine Steuerbefehle besitzt. So bleibt nur noch "GOTO" als letzte Rettung übrig. Da Sie allerdings mit ihrem CPC das ausgezeichnete Locomotive-BASIC erhalten haben, können Sie das "GOTO" weitgehend vermeiden.

Dieser Artikel soll allen "Weitsprung"-Geschädigten helfen, den "GOTO"-Befehl so einzusetzen, daß kein unentwirrbarer Spaghetti-Code entsteht.

Dies geschieht mit Hilfe der Regeln der Strukturierten Programmierung (im folgenden kurz : SP). Es stellt sich jetzt natürlich die Frage, was die so viel gerühmte SP eigentlich ist, und was deren Anwendung letztenendes bringt.

Unter dem Begriff Strukturierte Programmierung (SP) versteht man ein Verfahren, Programme so zu entwickeln, daß sie leicht durchschaubar und änderbar sind.

Wie alle guten Sachen hat aber auch die SP einen Haken. Besonders bei kurzen Programmen steigt der Zeitaufwand, wenn sie strukturiert programmiert werden.

Wenn man aber längere Programme erstellt, ist SP besonders effizient. Mit ihr können einzelne Routinen getrennt entwickelt und ausgetestet werden. Das bedeutet für Sie konkret, daß Sie zwar am Anfang etwas Zeit für die Planung eines Programmes opfern müssen, sich das Endprodukt aber sehen lassen kann. Sie erhalten ein leicht änderbares, übersichtliches Programm. Also schlechte Zeiten für "Drauflos"-Programmierer, denn das ist ein Tabu der SP. Und eben weil SP so eine gute Sache ist, will ich Sie nicht länger auf die Folter spannen.

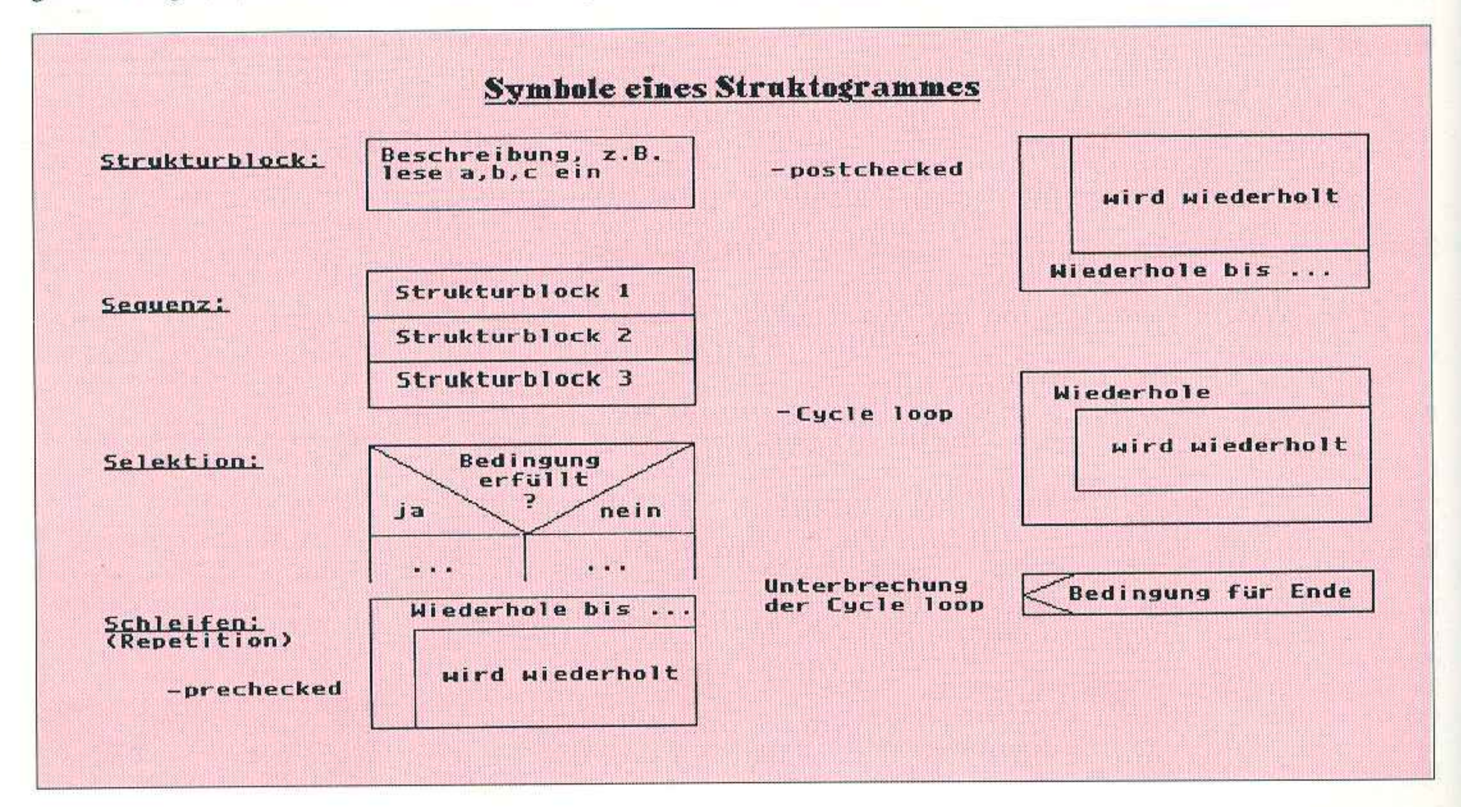
Der erste Schritt auf dem Weg zum strukturierten Programm, ist die Erstellung einer Analyse. Man stellt sich die Frage: Wie sieht überhaupt das Problem aus, das gelöst werden soll?

Will zum Beispiel Herr Hacker ein Programm schreiben, damit ihn jeden Morgen sein CPC weckt, so muß er im Sinne der SP erst einmal das Problem analysieren. Herr Hacker kommt nun zu folgendem Ergebnis: Das Programm WECKEN soll ihn an Wochentagen um 7.00 Uhr, und am Wochenende um 10.00 Uhr wecken, da er ausschlafen will. Mit diesen Informationen kann er aber noch kein Programm schreiben.

Er braucht erst eine Liste von Anweisungen, die das Problem lösen –den Ablaufplan. Er benutzt zum Entwurf des Ablaufplans ein Struktogramm. Das hat den Vorteil, daß er gar keine unstrukturierten Lösungen erstellen kann. Ein Struktogramm besteht aus beliebig vielen Strukturblöcken. Ein Strukturblock besteht aus einem Programmteil, der eine bestimmte einfache Aufgabe löst. Ein Strukturblock sollte nicht zu komplizierte Aufgaben enthalten, da dann die Klarheit und Übersicht verloren geht.

Am Anfang war ein Struktogramm

Aber nach welchen Regeln wird nun so ein Struktogramm erstellt? Es gibt mehrere Möglichkeiten, wie aus diesen vielen einzelnen Blöcken ein Programm wird, das eine Aufgabe löst. Erst einmal kann man diese Blöcke an-



Variablenliste des Programmes Funktionsplotter 1

Name	Typ	Funktionsbeschreibung
X	Real	X-Wert für die Funktionsgleichung
Y	Real	Y-Wert für den PLOT-Befehl
FN Y(X)	Funktion	Funktionsgleichung

Variablenliste des Programmes Funktionsplotter 2

Name	Typ	Funktionsbeschreibung
funktion$	String	Enthält die vom Benutzer gewünschte Funktionsgleichung.
X	Real	X-Wert für die Funktionsgleichung
Y	Real	Y-Wert für den PLOT-Befehl
FN Y(X)	Funktion	Funktionsgleichung

einanderreihen. Hierbei ist es sehr wichtig, daß ein Block genau einen Eingang und einen Ausgang besitzt (Sequenz). Das bedeutet: keine wilden GOTO-Sprünge in Strukturblöcke hinein und aus ihnen heraus. Aber keine Angst, auch wenn das GOTO in der SP etwas stiefmütterlich behandelt wird, es kommt dennoch nicht zu kurz.

Es gibt noch andere Schleifenarten, die sich nur dadurch unterscheiden, daß die Prüfung der Abbruchbedingung einmal am Anfang (prechecked loop), einmal am Ende (postchecked loop), und einmal in der Mitte liegt (Cycle loop).

Eine Schleife muß her

Nun gibt es in BASIC folgendes Problem: Der CPC kennt zwar noch die "prechecked" Schleife, in Form der Befehle "FOR, NEXT" und "WHILE, WEND", bei denen die Abbruchbedingung am Anfang geprüft wird, aber bei der "postchecked" Schleife läßt uns der BASIC-Interpreter in der Wüste stehen. Man behilft sich bei diesem Schleifentyp nun mit folgender Konstruktion:

```
10 REM Schleifenbeginn
20 REM große Arbeit
30 REM große Arbeit
...
100 IF NOT(Abbruchbedingung) THEN GOTO 10
```

Wie Sie sehen, ist der GOTO-Befehl doch für etwas gut, außer Unordnung zu stiften.

Außerdem darf sich der "GOTO"-Befehl noch beim Aufruf von Routinen zur Fehlerbehandlung und für Programmabbrüche sehen lassen. Ansonsten hat er "Hausverbot" in stilistisch guten Programmen.

GOTO ist doch erwünscht

Eine weitere Möglichkeit Ordnung zu schaffen besteht darin, häufig benötigte gleiche Programmteile in ein Unterprogramm zu packen, das dann bei Bedarf aufgerufen wird. In BASIC geschieht dies mit den Befehlen "GOSUB" und "RETURN". "GOSUB" sagt dem CPC, daß er zu einer anderen Zeilennummer springen soll. Aber im Gegensatz zu "GOTO" merkt sich der Computer die Position, die nach dem "GOSUB" kommt. Deshalb weiß er, wo er nach dem Ende des Unterprogrammaufrufs weitermachen soll. Dieses Ende wird durch den "RETURN"-Befehl gekennzeichnet.

In BASIC sieht das dann so aus:

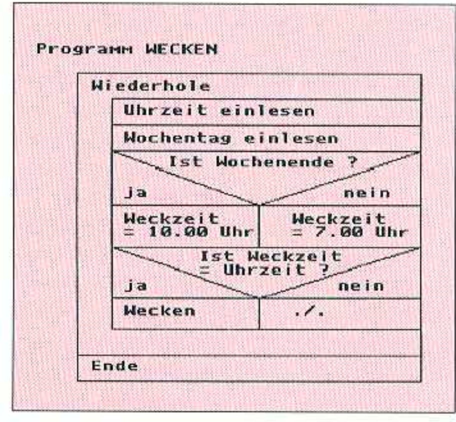
```
10 REM Hauptprogramm
20 GOSUB 100 : REM Hier kommt die Unterroutine
30 GOSUB 100 : REM nochmal
40 END : REM Das wars.
100 REM Unterroutine Schreibe "*"
110 PRINT "*";
120 RETURN : REM Jetzt gehts nach dem GOSUB weiter !
```

Man spart sich viel Arbeit und Speicherplatz, wenn man viele Unterroutinen benutzt. Allerdings steigern die Zeilennummern in BASIC nicht gerade die Verständlichkeit eines Unterprogrammaufrufes. Deshalb sollte man bei längeren Programmen eine Liste anlegen, in der die Funktion und die Zeilennummer einer Unterroutine vermerkt ist.

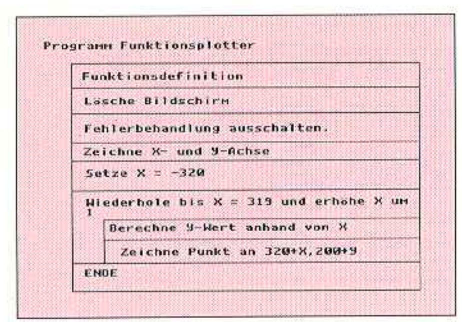
Wie war das doch gleich mit der "postchecked" Schleife? Sie ähnelt dem Anweisungspaar "REPEAT, UNTIL" in Pascal. Diese Befehle bewirken, daß die eingeschlossenen Anweisungen so lange ausgeführt werden, bis die Bedingung nach "UNTIL" erfüllt ist. Zum Beispiel rufen Sie so lange ihr Unterprogramm LESEN auf, bis sie von SP genug haben. Das ist auch eine REPEAT-UNTIL Schleife.

Schleifen besser ohne GOTO

Aufgrund der Nachahmung dieses Typs in BASIC muß die Abbruchbedingung verneint werden. Dann gibt es noch die Cycle-Schleife ohne Unterbrechung, die auch unter dem Namen "Endlosschleife" als Schreckgespenst durch den Kopf des leidgeplagten



Das Struktogramm des Weckprogramms



Die Darstellung des Programms – Funktionsplotter als Struktogramm

Computerfreaks geistert. Doch nur der Vollständigkeit halber will ich sie hier erwähnen, da sie sehr selten gebraucht wird. Wahrscheinlicher ist da schon die Verwendung der Cycle-Schleife mit Unterbrechung. Diese sieht aus, wie eine Endlosschleife, mit dem Unterschied daß sie irgendwo in der Mitte verlassen werden kann. In BASIC sieht diese Verwirrung stiftende Struktur so aus:

```
10 REM Schleifenbeginn
20 REM große Arbeit
30 REM schwere Arbeit
...
40 IF Bedingung THEN GOTO 100
...
90 GOTO 10 : REM Schleifenende
100 REM weiter gehts im Text ...
```

Wie unschwer zu erkennen ist, kann dieser Schleifentyp leicht auf den ersten Blick mit einer Endlosschleife verwechselt werden. Deshalb ist er nicht sehr günstig.

Nun gibt es aber auch noch einen Befehl, der auf den Namen "IF" hört und eine Verzweigung einleitet. Der Rest der Familie heißt "THEN" und "ELSE".

Diese logische Struktur findet sich auch in der SP unter dem hochwissenschaftlichen Namen "Selektion" wieder und erklärt sich abgesehen von dem Namen von selbst.

Verzweigung und Selektion

Zu erwähnen ist noch, daß sich im Sinne der SP die beiden Enden der Verzweigung weiter unten wieder treffen müssen. Andernfalls ist die IF-THEN Abfrage kein Strukturblock, denn der darf ja nur einen Eingang und einen Ausgang haben. Die Abbildung zeigt die den Befehlen entsprechenden Symbole. In der Abbildung finden Sie nun das hoffentlich etwas klarere Struktogramm zu Herrn Hackers Problem. Doch nun auf zu unserem ersten praktischen Beispiel:

Da Sie alle wahrscheinlich einen Wecker ihr eigen nennen, dürfte es ziemlich unnütz sein, selbigen zu programmieren. Wesentlich interresanter gestaltet sich da schon die Programmierung eines Funktionsplotters. Die Analyse ergibt, daß ein Programm zu schreiben ist, das eine Funktion zeichnet, die vom Benutzer eingegeben wird.

Zuerst erstellt man anhand dieser Informationen ein Struktogramm. Aus ihm geht hervor, daß die Fehlerbehandlung des Interpreters abgeschaltet wird. Das ermöglicht es Funktionen zu zeichnen, bei denen die Bereichsgrenzen des PLOT-Befehls überschritten werden. Damit ist die Programmplanung aber noch nicht abgeschlossen. Wie Sie vielleicht bemerkt haben, fehlt immer noch eine wichtige Sache. Neben dem Struktogramm, das den Ablauf des Programmes bechreibt, fehlt noch eine Variablen-Liste. Denn es ist ja noch nicht bekannt, welche Variablen benötigt werden und welchem Typ sie angehören sollen. Nachdem aber nun auch diese Liste erstellt worden ist, steht der Programmierung nichts mehr im Wege.

Das Problem der Funktionseingabe

Wenn Sie das Listing FPLOTT1.BAS ausprobiert haben, wird Sie warscheinlich die etwas unelegante "Routine" zur Eingabe der Funktion stören. Wenn zur Änderung der Funktionsgleichung jedesmal das Programm unterbrochen werden muß, ist das natürlich ärgerlich. Es geht aber auch anders. Viele Funktionsplotter behelfen sich mit MC-Routinen, die die BASIC-Zeile mit der Funktionsdefinition in das Programm einfügen, während es läuft. Das ist aber nicht nötig.

Für stolze Besitzer eines Diskettenlaufwerkes gibt es eine Alternative. Man erzeugt ein BASIC-Programm auf Diskette, das nur aus einer Zeile besteht. Diese Zeile enthält die vom Benutzer gewollte Funktionsgleichung. Wenn man nun diese Programmzeile mit dem Hauptprogramm verbindet, ändert sich die Funktionsgleichung. Der Knackpunkt dieser Idee ist das Verbinden von zwei Programmen. Wenn man es mit dem "MERGE"-Befehl versucht, erlebt man eine herbe Enttäuschung: Die Zeile wird zwar eingefügt, das Programm jedoch unterbrochen.

Speichern Sie bitte zuerst den Funktionsplotter ab, da wir ihn später nocheinmal brauchen. Zuerst müsen Sie die Zeile als ASCII-File abspeichern :

```
30 PRINT "HALLO":REM "MERGE"-Test
SAVE "ZEILE",A NEW
```

Dann können Sie das Hauptprogramm starten :

```
10 REM Hauptprogramm
20 MERGE "ZEILE"
30 REM Hier soll die Zeile hin !
```

Wichtig ist, daß Sie beim Eingeben genau dieselben Zeilennummern verwenden, da sonst die Zeile an die falsche Stelle geladen wird. Nach dem Start des Beispielprogrammes sieht das Listing so aus :

```
10 REM Hauptprogramm
20 MERGE "ZEILE"
30 PRINT "HALLO":REM "MERGE"-Test
```

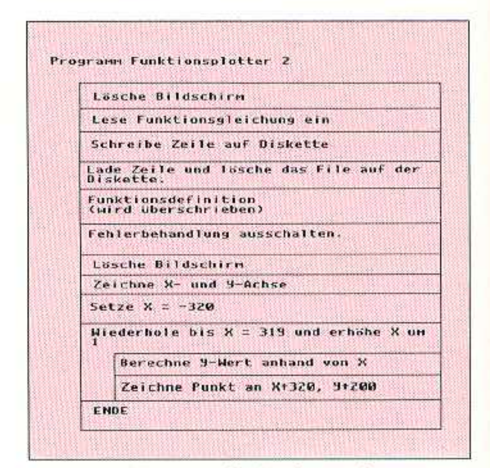
Also wurde die Zeile zwar nachgeladen, aber das Programm unterbrochen, da die Ausgabe "HALLO" fehlt. Ein Blick in das BASIC-Handbuch verrät, daß es noch zwei Befehle gibt, "CHAIN" und "CHAIN MERGE". "CHAIN" lädt ein Programm nach, löscht das alte und startet das neue. "CHAIN MERGE" ist eine Mischung aus beiden Befehlen, und genau das, was wir brauchen. Dieser Befehl hat aber noch Parameter. Dem Dateinamen folgt die Zeilennummer, an der das Programm fortgesetzt werden soll, dann kommt ein "DELETE"-Befehl. Er säubert den Bereich, in den der zu ladende Programmteil kommen soll Das verhindert Ärger mit den Resten eines Vorgängerprogrammes.

Die Lösung des Problems – CHAIN MERGE

Doch nun zu unserem erweiterten Beispielprogramm :

```
10 REM Hauptprogramm
20 CHAIN MERGE "ZEILE",30
30 REM Hier soll die Zeile hin !
```

Jetzt grüßt uns die nachgeladene Zeile, wie es sich gehört. Nach diesem Schema arbeitet auch der verbesserte Funktionsplotter, den Sie in Listing 2 abgedruckt finden. Das zugehörige Struktogramm zeigt die Abbildung.



Die Zweite Variante des Funktionsplotters

Worauf Sie in ihren Programmen achten müssen, wenn Sie diese Methode Benutzen wollen, ist, daß dabei nur der Inhalt der Variablen "überlebt". Die genaue Erläuterung finden Sie in ihrem BASIC-Handbuch unter den Befehlen "CHAIN" und "CHAIN MERGE".

Das Nachladen von Programmen kann man auch noch auf eine andere Weise nutzen. Vor allem bei großen Programmen ist es sinnvoll, nur häufig benötigte Routinen im Speicher zu halten. Das hat natürlich einen Nachteil: Besitzer eines Kassettenlaufwerks werden sich die Haare raufen, wenn sie nach jeder Eingabe den Programmteil "XYZ-0378" von Kassette laden müssen.

Ausnahmen bestätigen die Regel

Und auch Besitzer einer Floppy kann es auf die Dauer zermürben, wenn zu viel nachgeladen wird. Die Ausführungsgeschwindigkeit sinkt dabei ja rapide.

Und trotzdem: Es gibt Fälle bei denen es sinnvoll sein kann. Ein Beispiel wäre ein Programm, dessen Routinen nicht auf allen CPCs laufen. Dann kann man für jeden Computer eine Version der Routine bei Bedarf nachladen. Doch wie geht man am besten vor? Es sollte ja, wenn möglich, ein Programmteil nicht zweimal hintereinander nachgeladen werden.

Um den "CHAIN MERGE" kommt man nicht herum. Er muß immer dann kommen, wenn ein Programmteil nachgeladen werden soll. Aber um das doppelte Nachladen einer Routine zu vermeiden, kann man eine Variable einführen, in welcher der Name der zuletzt geladenen Routine steht. Das hört sich komplizierter an, als es ist. Unser Beispielprogramm von vorhin lautet abgeändert dann so:

Zuerst muß eine neue Unterroutine erzeugt werden, da ja die Zeilennummern nicht mehr stimmen :

```
1000 REM Name holen
1010 geladen$="SAGEHALLO"
1020 RETURN
1100 REM Routine SAGEHALLO
1110 PRINT "HALLO"
1120 RETURN
SAVE "HALLO",A
NEW
```

Jetzt kommt das neue Hauptprogramm:

```
10 REM Hauptprogramm
20 GOSUB 1000 : REM Name holen
30 IF geladen$<>"SAGEHALLO" THEN CHAIN
MERGE "HALLO",40,DELETE 1000-
40 GOSUB 1100 : REM Routine aufrufen
1000 REM nichts geladen
1010 geladen$="NICHTS"
1020 RETURN
1100 REM Routine tut nichts
1110 RETURN
```

Um keinen "Improper Argument" Fehler zu erhalten, muß in dem Bereich, den "DELETE" löschen soll, bereits etwas stehen.

Diesen Zweck erfüllen die Zeilen 1000 bis 1110. Außerdem sorgen sie dafür, daß dem String "geladen" ein Routinenname zugewiesen wird, und das Programm bei einem eventuellen Unterprogrammaufruf nicht im Nirwana landet.

Ich hoffe, daß Sie mit diesen Informationen den üblen Befehl "GOTO" nun aus ihren Listings weitgehend verbannen können. Viel Spaß beim Probieren.

Markus Kräutner, jg

Literatur:
[1] Karl Kurbel, Programm Entwicklung, Gabler Verlag Wiesbaden, 1990

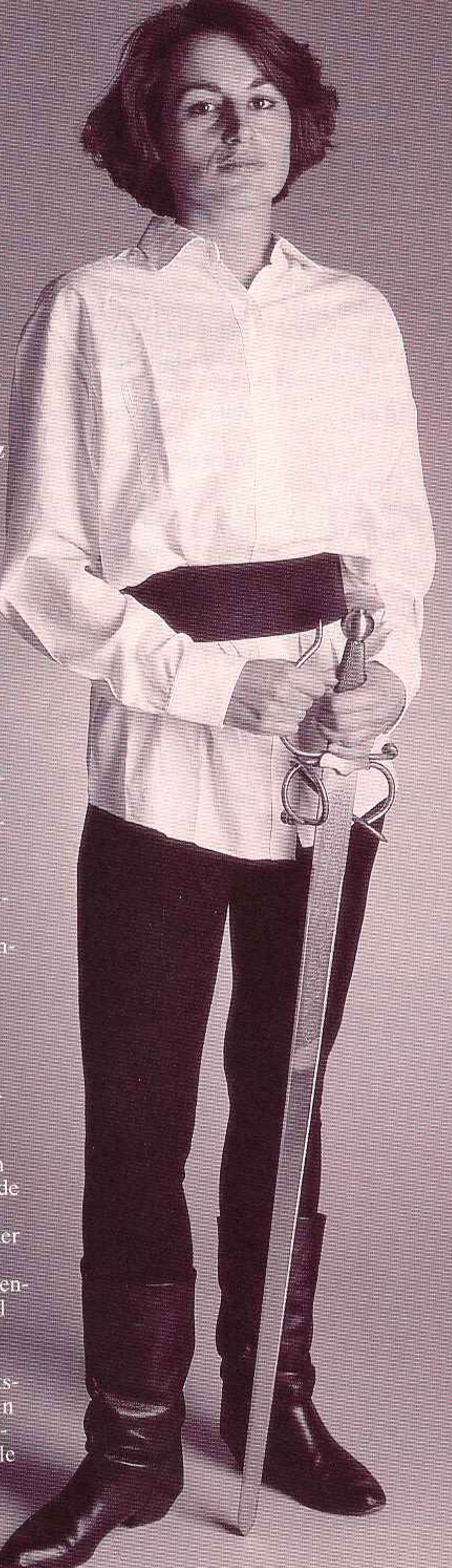

```
10 REM *****************************           [1365]
20 REM ***                         ***         [417]
30 REM ***    Funktionsplotter 1   ***         [1806]
40 REM ***(c)1991 Markus Kraeutner***          [2459]
45 REM *** & CPC International     ***         [1979]
48 REM ***                         ***         [417]
50 REM *****************************           [1365]
60 '                                           [117]
70 REM Funktionsdefinition                     [1758]
80 DEF FN y(x) = 0.0001*X*X*X                  [1805]
90 '                                           [117]
100 REM Loesche Bildschirm                     [2086]
110 MODE 2:INK 0,0:INK 1,26:BORDER 0           [3485]
120 '                                          [117]
130 REM Fehlerbehandlung aus                   [1784]
140 ON ERROR GOTO 330                          [1547]
150 '                                          [117]
170 REM Zeichne X- und Y-Achse                 [2074]
180 MOVE 0,200:DRAW 640,200                    [1361]
190 MOVE 320,0:DRAW 320,400                    [1904]
200 '                                          [117]
210 REM Wiederhole 640 mal                     [1607]
220 FOR X=-320 TO 319                          [855]
230   '                                        [117]
240   REM Berechne Y-Wert anhand von X         [2151]
250   Y = FN y(x)                              [1476]
260   '                                        [117]
270   REM Zeichne Punkt                        [1951]
280   PLOT X+320,Y+200,1                       [1073]
290   '                                        [117]
300 NEXT                                       [350]
310 END                                        [110]
320 '                                          [117]
330 REM Fehlerbehandlung : tue nichts          [2386]
340 RESUME NEXT                                [703]
```

```
10 REM *****************************           [1365]
20 REM ***                         ***         [417]
30 REM ***    Funktionsplotter 2   ***         [1751]
40 REM ***(c)1991 Markus Kraeutner***          [2459]
45 REM *** & CPC International     ***         [1979]
48 REM ***                         ***         [417]
50 REM *****************************           [1365]
60 '                                           [117]
70 REM Loesche Bildschirm                      [2086]
80 MODE 2:INK 0,0:INK 1,26:BORDER 0            [3485]
90 '                                           [117]
130 REM Lese Funktionsgleichung ein            [2512]
140 INPUT "Funktionsleichung : ",funktion$     [2826]
150 '                                          [117]
160 REM Schreibe Zeile auf Diskette            [871]
170 OPENOUT "Zeile"                            [1229]
180 PRINT #9,"260 DEF FN y(x) = "+funktion
$                                              [3597]
190 CLOSEOUT                                   [902]
200 '                                          [117]
210 REM Lade Zeile von Diskette in Zeile       [2243]
220 CHAIN MERGE "ZEILE",230                    [825]
230 A$="ZEILE":|ERA,@A$                        [1728]
240 '                                          [117]
250 REM Funktionsdefinition                    [1758]
260 DEF FN y(x) = 0.001*X*X-SIN(x*PI/180)*
0.01*x*x                                       [2621]
270 '                                          [117]
280 REM Fehlerbehandlung aus                   [1784]
290 ON ERROR GOTO 500                          [1345]
300 '                                          [117]
310 REM Loesche Bildschirm                     [2086]
320 CLS                                        [91]
330 '                                          [117]
340 REM Zeichne X- und Y-Achse                 [2074]
350 MOVE 0,200:DRAW 640,200                    [1361]
360 MOVE 320,0:DRAW 320,400                    [1904]
370 '                                          [117]
380 REM Wiederhole 640 mal                     [1607]
390 FOR X=-320 TO 319                          [855]
400   '                                        [117]
410   REM Berechne Y-Wert anhand von X         [2151]
420   Y = FN y(x)                              [1476]
430   '                                        [117]
440   REM Zeichne Punkt                        [1951]
450   PLOT X+320,Y+200,1                       [1073]
460   '                                        [117]
470 NEXT                                       [350]
480 END                                        [110]
490 '                                          [117]
500 REM Fehlerbehandlung : tue nichts          [2386]
510 RESUME NEXT                                [703]
```

Schwertgeschärfte Ohren

Serie: Der Weg zum eigenen Rollenspiel-System

Viele packende Stories schlummern noch verborgen in den hintersten Gehirnwindungen so manches "Bislang-noch-nicht-Programmierers" und warten darauf, in einem Computer-Rollenspiel zum Einsatz zu kommen. Wir verhelfen Ihnen zu Ihrem ganz persönlichen Abenteuer. Diesmal geht es um die ersten programmtechnischen Schritte: Ihr Spielprogramm lernt, Texteingaben zu verarbeiten.



Zu Anfang der 70er Jahre entwickelte der Amerikaner Gary Gygax aus der strategischen Militär-Simulation im Sandkasten-Diorama[2] ein eigenes Spielsystem mit komplexen Regeln. Ihm ging es nicht mehr um die möglichst realitätsnahe Nachbildung berühmter Schlachten, sondern um eine Darstellung fiktiver Fantasy-Welten.

Anstatt militärische Miniatur-Einheiten hin- und herzuschieben, übernahm nun jeder Mitspieler die Rolle eines Helden. Um unterschiedliche Charaktere zu kennzeichnen, wurden spezielle Eigenschaften nach Punkten bewertet. Klassische Ausprägungen sind etwa Stärke, Ausdauer und Charisma. Um für eine gute Story und die Einhaltung der Spielregeln zu garantieren, schuf Gygax die Position eines Spielleiters. Das ganze System nannte er dann "Dungeons & Dragons". Diese Urform aller heutigen Gesellschafts- wie auch Computer-Rollenspiele krankte jedoch besonders an dem Wust von Tabellen, Würfelorgien und Unmengen an Papierkram. Für fast jede Tätigkeit, jede Waffe, jede Situation gab es Besonderes zu beachten. Kaum ein Schritt konnte gewagt werden, ohne daß Spielleiter oder Mitspieler zum alles entscheidenden Würfel griffen.

Auch heute gibt es besonders von eingefleischten Rollenspielern bevorzugte Spiele, die nicht ohne einen Stapel Entscheidungstabellen und oft mit unterschiedlichen Würfelgarnituren gespielt werden.

Teilweise wird dabei auch ein Rechner in Gesellschafts-Rollenspiele integriert, um die Mitspieler vom Würfeln und dem endlosen Tabellenwerk zu entlasten. Entsprechende "Unterstützungsprogramme" für Gruppenspiele wie "Dungeons & Dragons" oder "Midgard" gibt es inzwischen auf verschiedenen Computersystemen; bisweilen sind sie ihrer kleinen Zielgruppe wegen

als Public-Domain-Software freigegeben.
Mit dem Aufkommen der ersten Heimcomputer entwickelten sich aber auch langsam die vom Teamgedanken losgelösten reinen Computer-Rollenspiele.

Do it yourself

In der letzten Folge unseres Rollenspiel-Kurses haben wir gezeigt, wie man eine gute Story entwickelt. Heute geht es darum, diese Story auf dem CPC zu verwirklichen.
Eine Möglichkeit, dem Rechner Züge, Situationen und Kommandos mitzuteilen, ist die Verwendung eines Parsers – also eines Programmoduls zur Textanalyse.
Die einfachsten Parser beschränken sich bei Dialogen mit dem Spieler auf die Verarbeitung simpler Zwei-Wort-Befehle. Derartige Abfragen lassen sich durch einfache "IF...THEN"-Konstrukte in BASIC realisieren.
Die nächste Leistungsstufe nach den simplen "SCHLAGE MONSTER"-Dialogen erreichen dann Parser, die auch schon komplexere Sätze mit Präpositionen und mehreren Objekten verstehen. Sätze wie "NIMM DIE KRONE UND BENUTZE DAS SCHWERT" sind ja auch schon fast Umgangssprache und können von besseren Algorithmen durchaus entschlüsselt werden.
Neben einer ausgefuchsten Satzanalyse gehört auch ein der Story entsprechender Wortschatz zu einem guten Abenteuer. Bei Rollenspielen fällt die Sprachbibliothek meist etwas bescheidener aus als bei echten Text- oder Grafikadventures. Geht es dort vielfach darum, eine vorgezeichnete Handlung durch Erraten und korrektes Formulieren der "richtigen" Reaktionen weiterlaufen zu lassen, so lebt das Rollenspiel meist von den spontanen Entscheidungen seiner Charaktere und kommt dafür mit weniger rätselhaften oder komplexen Tätigkeiten aus.

Satzanalyse

Die Eingabe von ganzen Sätzen ist beim BASIC der CPCs kein Problem. Über die "LINE INPUT"-Routine lassen sich ohne große Umstände Strings einlesen.

```
200 ...
210 LOCATE 1,1
220 LINE INPUT "Was nun? :"; t$
230 ...
```

Das nächste Problem ist die Wortseparierung innerhalb eines Satzes. Bevor wir nun Stück für Stück den Satz in einzelne Wörter zerlegen, empfiehlt es sich, alle Kleinbuchstaben in Großschrift umzuwandeln. Auch hierfür hält das CPC-BASIC eine komfortable Funktion bereit:

```
200 REM KLEIN -> GROSS
210 LOCATE 1,1
220 LINE INPUT "Was nun? :", t$
230 t$ = UPPER$(t$)
```

Bei Systemen, die "ß" unterstützen, muß dabei außerdem eine Umwandlung in "ss" vorgenommen werden, um groß und klein geschriebene Eingaben wirklich gleich behandeln zu können.
Der nächste Schritt ist die Zerlegung des Satzes in einzelne Wörter. Hier sollten wir dem Parser etwas "Intelligenz" spendieren und nicht nur auf Leerzeichen als Trennfuge zwischen Wörtern untersuchen. Die Ausschlußregel führt zu besserem Erfolg. Ist ein Zeichen kein Buchstabe, so postulieren wir, daß es sich wohl um ein Trennzeichen handeln muß.
Das Ergebnis dieser Überlegungen ist das Listingfragment, das wir im Kasten abgedruckt haben. Mit dieser kleinen Routine ist es möglich, die Anzahl aller in einem Satz vorkommenden Wörter zu ermitteln und die einzelnen Worte entsprechend abzutrennen.
Wie kommt man nun aber von einem solchen Sammelsurium einzelner Begriffsfragmente zu einer sinnvollen Satzauswertung?

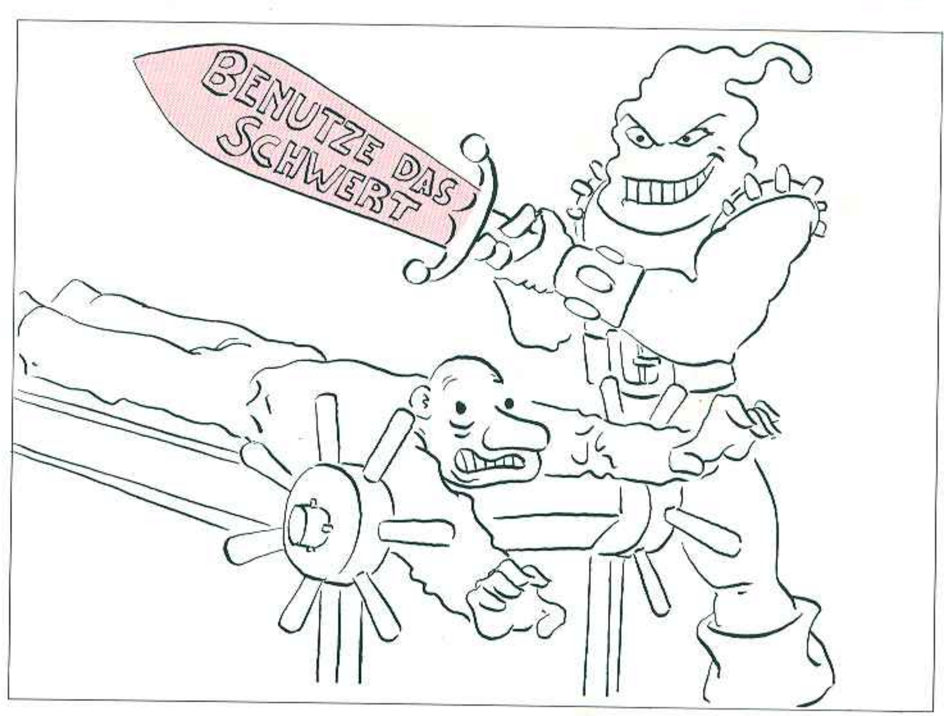
Erkenne meine Worte

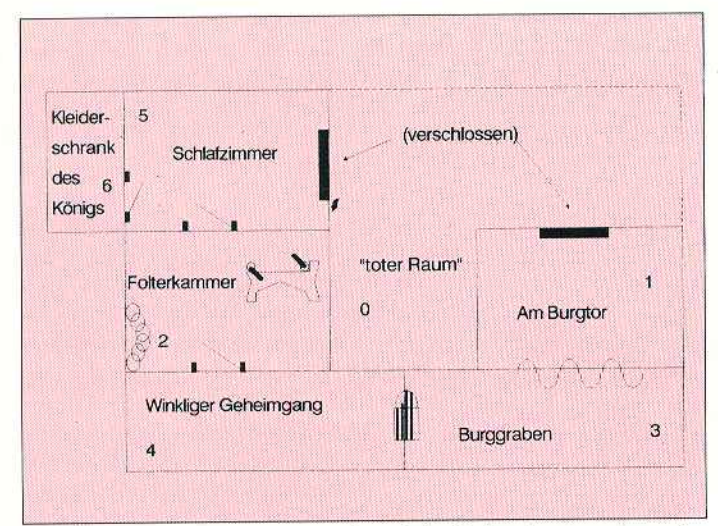
Der nächste Schritt besteht darin, die so als Einzelworte erkannten Begriffe mit dem im Programm verankerten Wortschatz zu vergleichen. Bei kleinen Wortschätzen lassen sich die verwendeten Begriffe noch ganz gut in DATA-Zeilen unterbringen. Im Zweifelsfall empfiehlt sich die externe Auslagerung auf Diskette, wobei möglichst immer Wortpakete oder Gruppen zur Auswertung zu laden sind. Würde man jedes Wort einzeln in den Arbeitsspeicher laden und dort mit dem Eingabewort vergleichen, entstünden für den Spieler unzumutbare Wartezeiten. Bei Rollenspielen und auch als allgemeine Unterstützung für Grafik-Adventures reicht in den meisten Fällen ein kleiner, über DATA-Zeilen ins Programm integrierter Wortschatz aus.

Um nun sinnvoll Worte zuordnen zu können, muß der jeweilige Wortschatz in einer fest definierten Struktur vorliegen. Hier gibt es sicher viele Möglichkeiten, wobei unser Vorschlag sich nur als Beispiel versteht. Wir teilen die archivierten Begriffe in fünf Gruppen auf:

- Objekte

Als Objekte werden alle Dinge im Programm bezeichnet, mit denen der Spieler agieren kann. Neben dem reinen Begriff sind unter Umständen auch Variablen für Position, Status (leere oder volle Flasche) sowie Gewicht und situationsbedingte Aktionen an das Wort zu knüpfen. So gibt es in fast jedem Rollenspiel zumindest eine Schatztruhe. Der in dieser Truhe liegende Ring sollte aber tunlichst erst sichtbar werden, wenn ein Heroe sich mit dem richtigen Schlüssel an dem Schloß der Truhe versucht hat, und nicht vorher. Mit





Die Skizze zu dem im Text verwendeten Beispiel für eine Lageplancodierung. Jede Verbindung zwischen zwei Räumen läßt sich als numerische Variable eines zweidimensionalen Feldes formulieren. Endet eine Bewegung in "Raum 0", so wird sie schlicht und einfach nicht ausgeführt



Das im Text erwähnte Programm "Eliza" hat im Laufe seiner Entwicklungsgeschichte schon seltsame Blüten getrieben: Hier taucht es als "eingebettetes" Spiel auf – es "läuft" auf einem simulierten Mini-PC innerhalb eines beliebten Krankenhaus-Simulationsspiels

einem auf gut Glück abgesetzten "Nimm Ring" bei geschlossener Truhe sollte man bestenfalls ein "Netter Versuch, du brichst dir die Fingernägel" erzielen.

Eine gute Idee ist es, für Standard-Begriffe wie "nimm", "sieh" oder "schlag" Synonyme und Abkürzungen zu erlauben. Selbst dem geduldigsten Abenteurer wird nach ein paar Stunden das kurze "ni Rüstung" lieber sein als die Langversion des Dialogs.

In Data-Zeilen könnte man eine derartige Struktur wie folgt aufbauen:

```
...
REM OBJEKTNAME, RAUM, SICHT, ZUGRIFF
DATA SCHWERT, 1, 2, 4
...
```

Die jeweiligen Zahlenwerte stehen dann enstprechend für verschiedene Räume und Objektzustände.

● Verben
Außer dem jeweiligen Verb sollte diese Struktur auch Namen mit jeweils gleicher Bedeutung enthalten. Es gibt tausend verschiedene Möglichkeiten, ein und dieselbe Aktion mit unterschiedlichen Worten darzustellen. Um hier langfristigen Spielspaß zu garantieren, sollte ein Gegner nicht nur nach "SCHLAG DAS MONSTER", sondern auch über "HAU DAS MONSTER" oder für die Feinschmecker mit "BEISS DAS MONSTER" zu überwinden sein.

Sehr elegant kann man solche Doppelbedeutungen verbinden, indem etwa für unser Beispiel "SCHLAG", "HAU" und "BEISS" mit einem stellvertretenden, identischen Wert ("Token") gekennzeichnet werden. Innerhalb des Programms wird dann nicht mehr mit den tatsächlichen Begriffen, sondern nur noch mit deren "Stellvertretern", also den Verben-Werten, gearbeitet. Diese haben den Vorteil, daß man damit rechnen kann. So lassen sich dann etwa bei Bedarf Gruppen nicht absolut gleicher, aber verwandter Befehle mit "Größer"- und "Kleiner"-Berechnungen zusammenfassen.

Für jedes Token müssen zusätzlich die erlaubten Objekte abgespeichert werden. Unsinnige Aktionen wie "BEISS DIE LUFT" können dann über einen simplen Standard-Weg mit einem "DAS GEHT NICHT" ohne große Logik beantwortet werden.

● Präpositionen
Präpositionen dienen im Deutschen dazu, räumliche oder zeitliche Bezüge anzudeuten. Klassische Präpositionen im Spiel sind "neben", "auf", "unter", "zwischen", "hinter" und "in".

Die Präpositionen sollten ähnlich wie Verben auch über einen Code angesprochen werden, um Doppelbedeutungen abzufangen und Objektbezüge herstellen zu können.

● Orte
Räume und Positionen spielen nur im Zusammenhang mit Verben oder Präpositionen eine Rolle. Zu ihrem Namen oder Bezeichner sollte man mögliche Richtungsangaben in Bezug bringen.

Man könnte beispielsweise ein Variablenfeld namens "RICHTUNG" definieren, welches die jeweils mögliche Marschrichtung speichert oder besser den als nächstes erreichbaren Ort vermerkt. Soll man von einer bestimmten Stelle aus nicht weiterkommen, weil etwa Felsen oder Abgründe den Weg versperren, wird für das Hindernis einfach ein "toter" Raum definiert. Beispiel: Setzen wir für die vier Windrichtungen jeweils Werte von "1" für Norden bis "4" für Westen. Der tote Raum bekommt den Wert "0". In einem zweidimensionalen Array "RICHTUNG(X,Y)" legen wir nun die geografischen Möglichkeiten fest. Von der Folterkammer (Raum 2) gelangt man ins Schlafzimmer (Raum 5), indem man nach Norden (Richtung 1) geht. Somit muß "RICHTUNG(2,1)" den Wert "5" enthalten und umgekehrt "RICHTUNG(5,3)" nach Süden (Richtung 3) hin den Wert "2" für die Folterkammer.
Die allgemeine Form des beschreibenden Arrays wäre also:

```
RICHTUNG(aktueller Raum, Marschrichtung) = nächster Raum
```

● Sonstiges
Unter "Ferner liefen" sollte man alle durch bisherige Gruppen nicht abgedeckten Begriffe speichern. Neben direkten Spielanweisungen wie etwa "Spielstand laden" oder "Spiel beenden" können das auch Kurzformen häufig benutzter Anweisungen wie beispielsweise "Norden" oder "Inventur" sein. Auch Funktionen wie "Flieh" und "Hilfe" haben hier ihren Platz.
Dieses Sammelbecken von Begriffen sollte in einer gesonderten Tabelle geführt werden.
Durch die Eigenart dieser Eingabe-Kommandos kann man noch ein besonderes Schmankerl ins Programm bringen, und zwar den sogenannten "Eliza"-Effekt.

"Eliza" ist inzwischen ein echter Software-Klassiker und führt auch in unserer heutigen computerverwöhnten Zeit immer noch zu Erstaunen und Stirnrunzeln. Dieses Primitiv-Programm, das es auf allen möglichen Computersystemen vom Großrechner bis zum Atari 800 gibt, täuscht nämlich mit sehr einfachen Mitteln Intelligenz vor. Es beantwortet mit Hilfe simpler Algorithmen jede Eingabe des Benutzers mit einer an bestimmten Bestandteilen aufgehängten Phrase, die wiederum gezielte Reaktionen weitgehend ausschließen muß. "Eliza" formt so einen tatsächlichen Monolog des Benutzers in einen Dialog um. Der Betrachter glaubt dann, sich mit dem Rechner zu unterhalten, obwohl im Prinzip nur die eigenen Worte durch hohle Phrasen verkleidet zurückgegeben werden.

"Eliza"-Programme kennen nämlich im Grunde genommen bloß Standard-Reaktionen auf eine kleine Liste bestimmter Stichworte. Die tatsächlichen Satzinhalte sind dem "Dummpatz" von Programm völlig unbekannt. Gibt man beispielsweise "Gib mir eine Taschenlampe" ein, so könnte ein einfacher "Eliza"-Algorithmus mit "Du hättest also gern so etwas wie eine Taschenlampe?" antworten. Das würde auch hervorragend bei "Hilfe", "Waffen" und anderen Objekten funktionieren.

Überraschungen

Nun kennt man ja seine Pappenheimer. Irgendwann entfährt auch dem friedlichsten Abenteurer ein deftiges Schimpfwort. So etwas gehört natürlich vom Parser abgefangen und witzig beantwortet. "Ausnahmen" dieser Art erhöhen den Spielspaß und täuschen oftmals über das tatsächlich schematische Vorgehen eines Programms hinweg.

Eine Überraschung der unangenehmen Art kommt auf Spieler wie auch Programmierer zu, wenn gleiche Begriffe mit unterschiedlichen Bedeutungen auftreten. Anstatt zu einer "kleinen" und "großen" Axt zu greifen, sollte man deshalb lieber gleich die Begriffe "Streitaxt" und "Beil" einführen.

Die eigentliche Auswertung und Prüfung der bekannten Worte kann auf vielerlei Weise vorgehen. Wir empfehlen, zuerst einmal auf Vollständigkeit des eingegebenen Satzes zu überprüfen. Er sollte also zumindest ein Schlüsselwort, ein Verb und ein Objekt enthalten. Ist dies nicht der Fall, bietet sich zum "Abwimmeln" der unbrauchbaren Eingabe noch die erwähnte "Eliza"-Methode an – oder einfach eine barsche Fehlermeldung.

Des weiteren sollte eine situationsabhängige Prüfung erfolgen. Unter diese Prüfung fallen Bedingungen wie "Verfügt der Spieler über genügend Geld, um eine Axt zu kaufen?" oder "Ist die Tür, die der Spieler öffnen möchte, schon mit dem Schlüssel geöffnet worden?". Auch Objekte, die schon einmal im Spiel auftauchten und jetzt jedoch unerreichbar in anderen Räumen schlummern, müssen berücksichtigt werden.

Schöne Aussichten

Zum Abschluß noch ein paar "goldene" Grundregeln für die eigene Parser-Programmierung. "Lebenswichtig" sind vor allen Dingen die Zerlegungs- und Interpretations-Prozeduren. Nach der Entwicklung der Story[1] gehört ein brauchbarer, eindeutig passender Wortschatz zu jedem guten Parser.

Auch wenn es für Ablaufgeschwindigkeit und Platzbedarf des Programms Besseres geben sollte: Die nötigen DATA-Zeilen gehören der Übersichtlichkeit halber ans Ende des Programms und sollten jeweils nicht zum Brechen vollgestopft werden. Es ist besser, ein paar DATA-Statements mehr zu setzen und dafür beim Ändern des DATA-Materials ein bißchen Platz für längere Stichworte in der Zeile zu behalten. Und geändert wird auf jeden Fall – mehr als einmal und an mehr als einem DATA-Statement. Verlassen Sie sich drauf.

Überlassen Sie auch die Abgrenzung einzelner DATA-Bereiche voneinander nicht der berühmten Zeilennummer in Verbindung mit RESTORE. Wesentlich professioneller ist eine jeweils für die einzelnen Wortgruppen spezifische Ende-Kennung in der DATA-Reihe, verbunden mit einer Ende-Abfrage beim READen. Eine solche Konstruktion erleichtert die Ergänzung des Wortschatzes (ist grundsätzlich fällig!) sowie die allgemeine Übersicht (sollte man nicht unterschätzen!).

Soviel also zur Verarbeitung von Texteingaben. Heutzutage wird allerdings für Rollenspiele nur selten eine reine Parsersteuerung verwendet. Meistens arbeitet ein guter Parser im Verbund mit grafisch gestützter Joystick- oder Maussteuerung, um die einzelnen Spielfiguren agieren zu lassen. Und um agieren zu können, braucht man ein Spielfeld – eine "map", um mal wieder einen Fachausdruck fallenzulassen. Unseren Heroen einen hübschen Lebensraum zu spendieren, soll denn auch Thema der nächsten Rollenspiel-Folge sein. Alsdann – schärfen Sie für diesmal mit dem Schwerte des Parsers die "Ohren" Ihres Programms. Beim nächsten Mal sollen dann auch die Augen zu ihrem Recht kommen.

Ulrich Schmitz/sz

Literatur:

[1] Ulrich Schmitz: Programmierte Traumwelten, CPC Amstrad International 10/11'91, S. 35-38

[2] Karlheinz Herzig: Der Sandkasten, Verlag Offene Worte, Bonn


```
100 ...
110 start = 1
120 worte = 1
130 ...
200 REM KLEIN -> GROSS
210 LOCATE 1,1
220 LINE INPUT "Was nun? :", t$
230 t$ = UPPER$(t$)
240 ...
300 REM WORT-SEPARIERER
310 FOR I = 1 TO LEN(t$)
320 IF MID$(t$,i,i) < "A" OR
MID$(t$,i,i) > "Z"
THEN GOTO 340
```

```
330 GOTO 390
340 REM WORTTRENNER ERKANNT
370 IF MID$(t$,i-1,i) >= "A" AND
MID$(t$,i-1,i) <= "Z" THEN
wort$(worte) = MID$(t$,start,((i-1)-start))
360 IF MID$(t$,i+1,i) >= "A" AND
MID$(t$,i+1,i) <= "Z" THEN
worte = worte + 1 :
start = i + 1
370 ...
390 NEXT I
400 PRINT "Insgesamt ";(worte);
" Worte im Satz gefunden."
410 ...
```



"Alkatrax"-Codierprogramm

Binärdateien leicht verschlüsselt

Das Thema Datenschutz ist schon seit den Kindertagen der Computer in aller Munde. Eine wichtige Maßnahme ist das Unlesbarmachen von Daten. Bisher konnte man nur auf dem CPC Datenfiles oder BASIC-Programme verschlüsseln. Wir stellen eine Möglichkeit vor, nun auch Maschinenprogramme vor fremdem Zugriff zu schützen.

Bisher gab es schon eine ganze Menge Codierprogramme, die jedoch alle bestimmte Nachteile hatten.

Entweder das Programm codiert mit einem Codewort, welches man leicht vergessen kann (außerdem muß man jedesmal wieder mit diesem Wort decodieren lassen), oder in einem Basic-Programm werden Zeilen durch Überlänge, falsche Zeilennummern und so weiter geschützt.

Eine weitere Möglichkeit eines Kopierschutzes ist, daß das Directory unsichtbar gemacht wird. Einfache Abhilfe: Diskmonitor, Directory Editor ... reinladen, Directory ändern, fertig. Alles wieder da. Aber warum eigene, wirkungslose und unbrauchbare Schutzarten ausdenken, wenn man nur bei den Profis schauen muß. Und genau das ist der mit Sicherheit wirkungsvollste Schutz, wie Sie gleich nach einer Erklärung erkennen können. Vorab noch eine kurze Information: Selbst der Entwickler dieses Schutzes bräuchte für ein normal geschütztes Programm vier bis zwölf Stunden (je nach Länge des Schutzes)!!! Nun folgen einige Informationen über diesen Kopierschutz und seine Arbeitsweise:

Jedem Computerbesitzer ist das Wort Kopierschutz ein bekannter Begriff (sofern er Originalsoftware benutzt und versucht hat, sie zu kopieren). Aber wie funktioniert ein Kopierschutz?

Die Grundlage ist eine mit einem Spezialformat formatierte Diskette. Dieses Format ist meist nur schwer zu kopieren. Das gesamte Programm ist dabei nicht als File abgespeichert, sondern direkt auf Diskette geschrieben. Um es einzulesen, wird eine Trackladeroutine aufgerufen, die das Programm von den Tracks in den Speicher liest. Wenn man diese Laderoutine, die oft sehr kompliziert ist, ungeschützt vorliegen hat, ist es meistens ein Leichtes (für Assembler-Kundige!), das Programm einlesen zu lassen und als File normal abzuspeichern. Auf diesem Prinzip beruht übrigens das Cracken von Originalsoftware:

Man versucht also an den Lader heranzukommen. Doch dies ist wirklich sehr schwer. Da die Programmierer natürlich nicht wollen, daß man leicht an diese Laderoutine kommt, verschlüsseln sie diese mit zirka 60 bis 300 Codierschleifen. Jetzt werden Sie fragen, was sind Codierschleifen und wie funktionieren sie ? So arbeiten diese Codierschleifen in Assembler :

Codierschleifen

Ein Register, meist HL, zeigt auf das erste Byte hinter dieser gerade "aktiven" Schleife. In BC oder DE steht die Anzahl der Bytes, die decodiert oder codiert werden sollen. Dann wird das Byte in HL eingelesen, verknüpft und wieder zurückgeschrieben. HL wird erhöht, DE beziehungsweise BC erniedrigt. Diese Prozedur wird so lange wiederholt, bis der Zähler (DE oder BC) gleich 0 ist.

Dann ist die nächste Schleife decodiert, und die darauffolgenden sind "etwas" decodiert. Bei dieser und den nächsten Schleifen geht's genauso weiter, wie eben gezeigt. Als Beispiel dient hier die Beispielroutine.

Nachdem die erste Schleife n mal (n = laenge) durchlaufen ist, steht die darauffolgende decodiert, also im Format wie die erste im Speicher und wird automatisch ausgeführt, weil ja nach dem letzten Befehl der ersten Schleife die zweite gleich beginnt. Und so geht das dann immer weiter, bis alle Schleifen beendet sind und das Programm richtig im Speicher steht.

Damit aber Programmcode decodiert werden kann, muß er vorher codiert werden. Man läßt den Vorgang also rückwärts ablaufen:

Combat School
Gryzor
Trantor - The Last Stormtrooper
Human Killing Machine
Thunder Blade
Sonic Boom
Die Arche des Captain Blood
Street Fighter
Last Ninja 2
Summer Games 1+2
Crackdown
Heath Wave (Compilation)
Arcade Muscles (Compilation)
10 MEGA Games (Compilation)

Diese Programme benutzen zum Beispiel den vorgestellten Kopierschutz.

```
begin  LD HL,start      ;Beginn der De - Codierung        BASIC : HL = start
       LD BC,laenge     ;laenge = ende - start + 1        BASIC : BC = laenge
loop   LD A,(HL)        ;Wert aus HL holen                BASIC : A = PEEK (HL)
       XOR wert         ;verknüpfen (wert:0-255)
       LD (HL),A        ;verknüpften Wert in HL schr.     BASIC : POKE HL, A
       INC HL           ;HL um eins erhöhen               BASIC : HL = HL + 1
       DEC BC           ;BC um eins erniedrigen           BASIC : BC = BC - 1
       LD A,B           ;Test, ob BC = 0                  BASIC : A = B
       OR C             ;Dazu wird B mit C geORt          BASIC : A = A OR C
       JR NZ,loop       ;Nicht 0 ? Dann nochmal           BASIC : IF A<>0 THEN loop
```

Beispiel für die (De)Codierung eines Binärfiles. Anschließend folgt das Programm.

Zuerst beginnt man mit der letzten Schleife bei der Codierung, dann bei der vorletzten und so weiter bis hin zur ersten. Nun steht alles geXORt im Speicher. Das Schwierige am Cracken, dem Hacken der Schleifen, ist aber, daß es so unwahrscheinlich viele Sorten davon gibt. Es werden insgesamt zirka fünfzig Typen verwendet. Aber wir wollen ja nicht cracken, weil wir keine Raubkopierer sind. Wozu aber dann die ganzen Erklärungen ?

- weil man so die Funktionsweise eines Kopierschutzes durchschauen kann,
- weil man so einsehen kann, daß dies der sicherste Kopierschutz überhaupt sein muß und
- weil dadurch die Funktionsweise meines Codierprogrammes anschaulich wird!

Funktionsweise des Programmes:

Nach dem Start werden Sie nach dem Namen des zu codierenden Programmes (nur Binärprogramme) und dessen Startadresse gefragt. Die Originalstartadresse muß hierbei unbedingt beibehalten werden! Dann kommt eine Abfrage nach der Anzahl der Schleifen. Geben Sie hier einen Wert zwischen 1 und X ein. X sollte nicht zu groß sein, da sonst der Speicherplatz nicht reichen würde. Aber vierzig Schleifen wären sicher schon genug.

Und dann geht's auch schon los! Das Programm wird codiert. Je nach Länge und Anzahl der Schleifen dauert dies einige Zeit; bei einer Länge von 4096 Bytes und einer Schleifenanzahl von vierzig zum Beispiel 45 Sekunden. Danach werden Sie aufgefordert, eine Taste zu drücken. Das fertige Produkt wird als CODE.BIN abgespeichert. Falls keine Abspeicherung gewünscht ist, drücken Sie einfach zweimal die Escape-Taste. Das folgende kurze Programm veranschaulicht eine Codierung:

```
FOR X=&4000 TO &5000:POKE X,&C9:NEXT
```

Danach das Codierprogramm starten, bei NAME bitte RETURN drücken, damit kein Programm geladen wird. Als Startadresse &4000, als Endadresse &5000. Danach merken Sie sich bitte die neue, vom Programm angegebene Startadresse. Danach sehen Sie sich den Speicher von &4000 bis &5000 wie folgt an:

```
FOR X=&4000 TO &5000:? HEX$(PEEK(X))"
";:NEXT
```

Jetzt müßten lauter verschiedene Werte ausgegeben werden. Das sind die codierten Bytes. Dann rufen Sie die neue Startadresse auf, die Sie sich gemerkt haben, mit: `CALL neue Startadresse` . Jetzt schauen wir nochmal den Speicher an:

```
FOR X=&4000 TO &5000:? HEX$(PEEK(X))"
";:NEXT
```

Der alte Wert &C9 steht jetzt wieder da. Wenn Sie ein Programm schützen, dann bitte nur ein Programm, welches selbständig lauffähig ist, also mit *CALL adresse* aufgerufen werden kann, und keine Grafik oder etwas in dieser Richtung. Viel Spaß beim Verschlüsseln.

Anmerkung: Die Erklärungen verlangen eine gewisse Vorkenntnis in der Programmiersprache Assembler. Aber auch Assembler-Unkundige können das Prinzip verstehen, nicht aber die Erklärung der Codierschleifen, was für die Benutzung des Programmes nicht erforderlich ist!

Klaus Meffert/rs

```
10 'CODIER.LDR                                   [777]
20 'Generiert PROGRAMM.BIN fuer das              [2821]
30 'Alkatraz Protection System                   [2837]
40 '(C) 1991 Klaus Meffert                       [435]
50 '     & CPC Internationl                      [1846]
60 '                                             [117]
70 FOR i=&AF00 TO &AFBF                          [1282]
80 READ a$:w=VAL("&H"+a$)                        [880]
90 s=s+w:POKE i,w:NEXT                           [1039]
100 IF s<> 21056 THEN PRINT"Fehler":END          [2578]
110 SAVE"codier.bin",b,&AF00,&C0:END             [2169]
120 DATA 3E,08,32,2C,AF,21,F0,3F                 [903]
130 DATA 01,10,02,C5,7E,57,3A,2C                 [856]
140 DATA AF,5F,06,09,3C,FE,80,C2                 [1399]
150 DATA 1C,AF,D6,80,10,F6,32,2C                 [1607]
160 DATA AF,7B,AA,77,23,C1,0B,78                 [961]
170 DATA B1,20,E0,C9,18,01,00,00                 [975]
180 DATA 16,00,21,00,00,01,00,00                 [1422]
190 DATA 5E,7B,AA,77,53,0B,23,78                 [1593]
200 DATA B1,C2,38,AF,C9,3E,09,32                 [1319]
210 DATA F0,AF,21,30,40,01,10,00                 [1439]
220 DATA 3A,F0,AF,C5,57,06,0D,3C                 [1530]
230 DATA FE,80,C2,5F,AF,D6,80,10                 [1722]
240 DATA F6,32,F0,AF,7E,AA,57,3A                 [1499]
250 DATA F1,AF,AA,57,3A,F2,AF,AA                 [1124]
260 DATA 77,00,00,23,C1,0B,78,B1                 [1965]
270 DATA C2,50,AF,C9,3E,0A,32,2D                 [350]
280 DATA AF,21,45,87,01,02,02,FD                 [1586]
290 DATA 21,FF,3F,C5,3A,2D,AF,FD                 [1724]
300 DATA AE,00,AC,AD,FD,77,00,0B                 [1150]
310 DATA 2B,FD,23,3A,2D,AF,06,0F                 [1815]
320 DATA 3C,FE,80,C2,A8,AF,D6,80                 [1378]
330 DATA 10,F6,32,2D,AF,C1,0B,78                 [751]
340 DATA B1,C2,8B,AF,C9,FF,FF,FF                 [1569]
350 DATA 00,00,00,00,FF,FF,FF,FF                 [1667]
```

```
10 '************************************         [1383]
20 '*Special Alkatraz Protection System*         [2073]
30 '*Version 2.0 (c) 1991 Klaus Meffert*         [1926]
40 '*              & CPC International *         [966]
50 '************************************         [1383]
60 'Dieses Programm erstellt einen Kopier-       [3011]
70 'schutz, wie er bei komerzieller Soft-        [3307]
80 'ware verwendet wird.                         [1653]
90 '                                             [117]
100 'Vorbereitung                                [643]
110 LOAD"codier.bin",&AF00                       [1692]
120 zyk=6:var=0:stufe=0                          [1965]
130 'Kopfzeile ausgeben                          [1933]
140 MODE 2                                       [513]
150 PRINT STRING$(80,"*");                       [1398]
160 PRINT"    ALKATRAZ PROTECTION SYSTEM         [5873]
(c) 1991 Klaus Meffert & CPC International
"
170 PRINT STRING$(80,"*")                        [1446]
180 'Eingaben                                    [1013]
190 INPUT"Programmname : ",name$                 [2404]
200 IF name$="" THEN 230                         [393]
210 INPUT"Startadresse : ",adr                   [2868]
220 MEMORY adr-1:LOAD ""+name$,adr               [349]
230 INPUT"Wieviele Schleifen : ",loops           [3421]
240 'Variablen entsprechend den Eingaben d       [3262]
imensionieren
250 DIM a$(40),art(loops),lo(loops-1),hi(l       [4750]
oops-1),lo2(loops-1),hi2(loops-1)
260 DIM zy(loops-1),wert(loops-1),typ(loop       [5846]
s-1),lo3(loops-1),hi3(loops-1)
270 PRINT"Welcher Bereich soll codiert wer       [2969]
den ?
280 INPUT"Von : ",anfang                         [2117]
290 INPUT"Bis : ",ende                           [768]
```

```
300 FOR x=1 TO loops                                [1525]
310  art(x)=INT(RND*5)+1                            [1243]
320 NEXT                                            [350]
330 'Benoetigten Speicher berechnen                 [3036]
340 aut=1:space=anfang-1                            [2353]
350 laenge=ende-space+1                             [1235]
360 IF laenge<0 THEN laenge=laenge+65536            [2585]
370 LOCATE 1,6:FOR x=1 TO 6:PRINT SPACE$(8 [2640]
0);:NEXT
380 PRINT:FOR x=1 TO loops                          [2121]
390  IF art(x)=1 THEN space=space-18                [2735]
400  IF art(x)=2 THEN space=space-16                [1335]
410  IF art(x)=3 THEN space=space-25                [1905]
420  IF art(x)=4 THEN space=space-28                [1208]
430 NEXT                                            [350]
440 space=space-6:begin=space                       [1160]
450 IF space<&1C50+loops*50 THEN LOCATE 1, [9519]
25:PRINT"          Achtung - Zu wenig Spe
icher - Bitte eine Taste druecken ! ":CALL
 &BB06:RUN 120
460 laenge=ende-space+1                             [1235]
470 IF laenge<0 THEN laenge=laenge+65536            [2585]
480 'Dateiinformationen ausgeben                    [1309]
490 LOCATE 55,5:PRINT"Start  :"HEX$(space, [8818]
4):LOCATE 55,6:PRINT"Laenge  :"HEX$(laenge+
57,4):LOCATE 55,7:PRINT"Ende    :"HEX$(ende
+57,4)
500 gesamt=laenge+space                             [1472]
510 GOSUB 1350:RESTORE 1450:FOR x=1 TO 7:R [5241]
EAD a$:POKE space,VAL("&"+a$)
520 space=space+1:NEXT                              [872]
530 GOSUB 1350:LOCATE 1,12:PRINT"Loop Nr.: [5682]
":FOR zaw=1 TO loops:LOCATE 10,12:PRINT za
w
540 ON art(zaw) GOSUB 760,890,1020,1170             [1566]
550 NEXT                                            [350]
560 GOSUB 1350:RESTORE 1470:spa=ende+1              [2885]
570 READ a$:FOR x=1 TO LEN(a$)                      [717]
580 POKE spa,ASC(MID$(a$,x,1)):spa=spa+1:N [2091]
EXT
590 'Programm codieren                              [1190]
600 GOSUB 1350:PRINT:PRINT:PRINT"Codiere   [5872]
...... Bitte warten !!!!"
610 FOR x=var-1 TO 0 STEP -1:ON typ(x) GOS [3063]
UB 680,700,720,740
620 NEXT                                            [350]
630 LOCATE 1,24:PRINT"Neue Startadresse :  [4251]
"HEX$(begin)
640 PRINT"           Bitte die Sicherungsdis [7265]
kette einlegen und eine Taste druecken":CA
LL &BB06
650 SAVE"code",b,begin,laenge+57,start              [2678]
660 END                                             [110]
670 'Codieren mit Typ 1                             [657]
680 POKE &AF33,lo(x):POKE &AF34,hi(x):POKE [5878]
 &AF36,lo2(x):POKE &AF37,hi2(x):POKE &AF31
,wert(x):CALL &AF30:RETURN
690 'Codieren mit Typ 2                             [664]
700 POKE &AF06,lo(x):POKE &AF07,hi(x):POKE [8181]
 &AF09,lo2(x):POKE &AF0A,hi2(x):POKE &AF01
,zy(x):CALL &AF00:RETURN
710 'Codieren mit Typ 3                             [663]
720 POKE &AF4B,lo(x):POKE &AF4C,hi(x):POKE [9090]
 &AF4E,lo2(x):POKE &AF4F,hi2(x):POKE &AF46
,zy(x):POKE &AFF1,lo3(x):POKE &AFF2,hi3(x)
:CALL &AF45:RETURN
730 'Codieren mit Typ 4                             [670]
740 POKE &AF7D,zy(x):POKE &AF89,lo(x):POKE [8276]
 &AF8A,hi(x):POKE &AF85,lo2(x):POKE &AF86,
hi2(x):POKE &AF82,lo3(x):POKE &AF83,hi3(x)
:CALL &AF7C:RETURN
750 END                                             [110]
760 'Typ 1                                          [82]
770 lo=0:hi=0:wert=INT(RND*65)+1:wert(var) [4791]
=wert:merk=space:RESTORE 1370
780 FOR x=1 TO 18:READ a$(x):POKE space,VA [4857]
L("&"+a$(x)):space=space+1:NEXT
790    POKE merk+7,wert                             [972]
800 space=space-1:hi=INT(space/256):lo=spa [4530]
ce-hi*256:IF hi<0 THEN hi=256+hi
810    POKE merk+1,lo:POKE merk+2,hi                [1448]
820 lo(var)=lo:hi(var)=hi                           [1808]
830 zahl=ende-space+1:hi=INT(zahl/256):lo= [5869]
zahl-hi*256:IF hi<0 THEN hi=256+hi
840    POKE merk+4,lo:POKE merk+5,hi                [933]
850 lo2(var)=lo:hi2(var)=hi:zyk=zyk+(ende- [6189]
space+1)*9:zyk=zyk+3:space=space+1
860    GOSUB 1340                                   [974]
870 typ(var)=1:var=var+1                            [491]
880 RETURN                                          [555]
890 'Typ 2                                          [89]
900 zyk=zyk+2:GOSUB 1340:zy(var)=zyk                [1896]
910 rreg=1:lo=0:hi=0:merk=space:RESTORE 13 [4338]
90:FOR x=1 TO 16:READ a$(x)
920    POKE space,VAL("&"+a$(x)):space=spac [3056]
e+1:NEXT
930 space=space-1:hi=INT(space/256):lo=spa [4530]
ce-hi*256:IF hi<0 THEN hi=256+hi
940    POKE merk+1,lo:POKE merk+2,hi                [1448]
950 lo(var)=lo:hi(var)=hi                           [1808]
960 zahl=ende-space+1:hi=INT(zahl/256):lo= [5869]
zahl-hi*256:IF hi<0 THEN hi=256+hi
970    POKE merk+4,lo:POKE merk+5,hi                [933]
980 lo2(var)=lo:hi2(var)=hi:zyk=zyk+(ende- [4565]
space+1)*9:space=space+1
990    GOSUB 1340                                   [974]
1000 typ(var)=2:var=var+1                           [1314]
1010 RETURN                                         [555]
1020 'Typ 3                                         [88]
1030 zyk=zyk+5:GOSUB 1340:zy(var)=zyk:merk [4195]
=space:RESTORE 1410:FOR x=1 TO 25
1040 READ a$(x):POKE space,VAL("&"+a$(x)): [2852]
space=space+1:NEXT
1050 hi=INT((space-16)/256):lo=(space-16)- [2681]
hi*256:IF hi<0 THEN hi=256+hi
1060    POKE merk+7,lo:POKE merk+8,hi               [1176]
1070 lo3(var)=lo:hi3(var)=hi                        [1276]
1080 space=space-1:hi=INT(space/256):lo=sp [4530]
ace-hi*256:IF hi<0 THEN hi=256+hi
1090    POKE merk+1,lo:POKE merk+2,hi               [1448]
1100 lo(var)=lo:hi(var)=hi                          [1808]
1110 zahl=ende-space+1:hi=INT(zahl/256):lo [5869]
=zahl-hi*256:IF hi<0 THEN hi=256+hi
1120    POKE merk+4,lo:POKE merk+5,hi               [933]
1130 lo2(var)=lo:hi2(var)=hi:zyk=zyk+(ende [4306]
-space+1)*13:space=space+1
1140    GOSUB 1340                                  [974]
1150 typ(var)=3:var=var+1                           [556]
1160 RETURN                                         [555]
1170 'Typ 4                                         [95]
1180 zyk=zyk+4:GOSUB 1340:zy(var)=zyk               [2208]
1190 lo=0:hi=0:merk=space:RESTORE 1440:FOR [2183]
 x=1 TO 28:READ a$
1200    POKE space,VAL("&"+a$):space=space+ [1634]
1:NEXT
1210 space=space-1:hi=INT(space/256):lo=sp [4530]
ace-hi*256:IF hi<0 THEN hi=256+hi
1220    POKE merk+5,lo:POKE merk+6,hi               [2271]
1230 lo(var)=lo:hi(var)=hi                          [1808]
1240 zahl=ende-space+1:hi=INT(zahl/256):lo [5869]
=zahl-hi*256:IF hi<0 THEN hi=256+hi
1250    POKE merk+1,lo:POKE merk+2,hi               [1448]
1260 lo2(var)=lo:hi2(var)=hi                        [1812]
1270 RANDOMIZE timer:lo3(var)=INT(RND*256) [4785]
:POKE merk+8,lo3(var)
1280 hi3(var)=INT(RND*256):POKE merk+9,hi3 [4182]
(var)
1290 zyk=zyk+(ende-space+1)*15:space=space [2230]
+1
1300    GOSUB 1340                                  [974]
1310 typ(var)=4:var=var+1                           [1161]
1320 RETURN                                         [555]
1330 '                                              [117]
1340 za=zyk:za=INT(za/128):zyk=zyk-za*128: [3626]
RETURN
1350 stufe=stufe+1:LOCATE 55,9:PRINT"Stufe [3679]
 : "stufe:RETURN
1360 'DATA-FIELD                                    [411]
1370 '1.Typ                                         [95]
1380 DATA 21,00,00,01,00,00,16,00,7e,aa,57 [3233]
,77,23,0b,78,b1,20,f6
1390 '2.Typ                                         [719]
1400 DATA 21,00,00,11,00,00,ed,5f,ae,77,1b [1685]
,7a,b3,23,20,f6
1410 '3.Typ                                         [190]
1420 DATA 21,00,00,01,00,00,cd,00,00,dd,e1 [4573]
,ed,5f,dd,ac,ae,dd,ad,77,0b,79,23,b0,20,f2
1430 '4.Typ                                         [303]
1440 DATA 01,00,00,fd,21,00,00,21,00,00,ed [4758]
,5f,fd,ae,00,ac,ad,fd,77,00,0b,2b,fd,23,78
,b1,20,ee
1450 'INTRODUCTION                                  [1108]
1460 DATA f3,ed,56,3e,04,ed,4f                      [1371]
1470 DATA "Special Alkatraz Protection Sys [8821]
tem (c)1991 Klaus Meffert & CPC Internatio
nal"
1480 DATA @                                         [215]
```

Gewitzt und fix mit RSX

BASIC-Programme in Assembler aufmöbeln

Das bunte Herbstlaub liegt regendurchtränkt auf den Straßen, die Abende werden unbarmherzig länger, und der CPC liegt willig auf dem Schreibtisch. Wer bis zum nächsten Frühling fit in Assembler sein möchte und dem Locomotive-BASIC des CPC mit selbstgestrickten Befehlserweiterungen auf die Sprünge helfen will, sollte auf keinen Fall diese Folge des Assembler-Kurses überblättern.

Bei Fernseh-Schnulzen beginnen einzelne Folgen oftmals mit "Was bisher geschah". Anders bei uns: Jeder Beitrag unserer "Assembler-Ecke" bildet eine eigene Einheit, so daß man jederzeit auf den Zug aufspringen kann. Diesmal heißt das Motto "RSX-Einbindung".

Was benötigt man, um den Zutritt zur Welt blitzeschneller und kompakter Maschinenspracheprogramme zu bekommen? Frohe Kunde: Sie brauchen vorerst nur den Standard-Editor des BASIC, das in Ihrem CPC eingebaut ist, sowie einen Assembler (etwa den CPC-Assembler V 2.0 aus Heft 8/9'91) für Ihre Entdeckungsreise.

Übrigens: Die Beispiel-Listings, die wir in unsere Artikel eingebaut haben, sind auf den gerade erwähnten CPC-Assembler zugeschnitten. Für andere Assembler, die über einen eigenen Editor verfügen, ist in jeder Zeile das Kommentarzeichen zu entfernen. Außerdem kann es je nach Assembler erforderlich sein, die Zeilennummern zu eliminieren. Näheres hierzu verrät Ihnen das Handbuch Ihres Assemblers.

Offenes BASIC

Das Erweitern des Basic-Befehlssatzes ist erfreulicherweise schon von den CPC-Entwicklern bei Amstrad vorgesehen worden. Nicht aus Nächstenliebe den kreativen CPClern gegenüber, sondern um selbst schnell und einfach Routinen für das Betriebssystem einbinden zu können. All die nützlichen und oft verwendeten Befehle mit einem sogenannten Pipe-Zeichen am Anfang sind Erweiterungsbefehle.

Das "¦" teilt dem System mit, daß der darauf folgende Befehl als Erweiterungsbefehl zu behandeln ist.

Um dem Kind einen Namen zu geben, bezeichnet man diese Art der Befehlsimplementation als "RSX". Das steht dabei für "Resident System Extension", was soviel bedeutet wie "residente Systemerweiterung" und damit den Nagel auf den Kopf trifft. Ursprünglich war diese Einrichtung für feste Systemerweiterungen in Form von "Expansions-ROMs" gedacht. Die von Haus aus vorhandenen RSXe wie etwa "¦DIR", "¦A", "¦REN", "¦ERA" usw. sind ja auch tatsächlich als interne Kommandos im ROM verankert. Dem System selbst ist es jedoch völlig egal, ob solche Befehle nun im RAM oder ROM auf Abruf warten.

Dem Einbinden eigener Befehlserweiterungen steht also nichts im Wege.

First Step

Aus dem BASIC kennt man den POKE-Befehl. POKE schreibt einen Wert von 0 bis maximal 255 in eine beliebige Speicherstelle. PEEK liest entsprechend einen Wert aus einer Speicherstelle aus. Wir wissen[2], daß ein Wert von 0 bis 255 genau im Darstellungsbereich eines Bytes liegt. Für die weitere Arbeit in Assembler wie auch in Basic ist es aber sehr oft nötig, einen 16-Bit-Wert in eine Speicherstelle zu bekommen.

Glücklicherweise sind wir ja angehende Assemblianer und kennen dementsprechend keine Grenzen mehr. Machen wir uns also einen eigenen POKE-Befehl für Werte im Bereich 0 bis hexadezimal FFFF. Neben dem Spaß am Hacken dürfen wir uns auch als Namensgeber betätigen. Über Geschmack läßt sich ja bekanntlich nicht streiten – es erleichtert allerdings die Arbeit mit solchen Befehlen ungemein, wenn sie "sprechend" sind. Andererseits macht es auch viel Eindruck auf Freund oder Freundin, wenn plötzlich ein "¦Bianca, &0000, &6666" zum völligen Systemabsturz führt.

Da es sich bei unserer Routine praktisch um einen "doppelten" POKE handelt, setzen wir als Beispiel einfach das Präfix "D" vor den klassischen POKE und erhalten einen wunderbar sprechenden DPOKE-Bezeichner. Die Parameterübergabe erfolgt bei unserem neuen Befehl analog zum klassischen POKE.

Begriffsklärung für den Einsteiger

RAM:
"Random Access Memory", auch "flüchtiger Speicher" genannt. Der Arbeitsspeicher, der für die vorübergehende Speicherung von Programmen und Daten verwendet wird, besteht aus RAM-Bausteinen. Diese verlieren die in ihnen gespeicherte Information, sowie die Stromversorgung unterbrochen wird. Bei Taschencomputern werden bisweilen stromsparende RAM-Typen eingesetzt, die ihren Inhalt dank einer Batterieversorgung über längere Zeit aufrechterhalten können. Der Zugriff auf RAM-Bausteine kann in der Regel sehr schnell geschehen.

ROM:
"Read Only Memory", also Nur-Lese-Speicher, auch "Festspeicher" genannt. Speicherbausteine, die nach ihrer industriellen Einmal-Programmierung weder gelöscht noch neu beschrieben werden können. Sie enthalten Programme, die zum festen Einbau in Rechner oder andere prozessorgesteuerte Geräte gedacht sind. So verfügt beispielsweise der CPC ebenso über ROMs wie jeder handelsübliche CD-Spieler. Auf ROM gespeicherte Programme lassen sich nur durch Austausch des entsprechenden Bausteins "updaten" (ändern). Beim CPC ist außer dem Betriebssystem AMSDOS auch das Locomotive-BASIC fest im ROM verankert. Für das Lesen von Daten braucht ein Computer beim ROM im allgemeinen wesentlich länger als beim RAM. Dem CPC ist dies allerdings egal; er ist ohnehin nicht der Schnellste und kann die Möglichkeiten des flotteren RAM-Zugriffs nicht nutzen.

EPROM:
"Erasable Programmable Read Only Memory", zu deutsch "Löschbarer, programmierbarer Nur-Lese-Speicher". Eine Art Kompromiß zwischen dem kurzatmigen RAM und dem starren ROM. Ein EPROM-Baustein läßt sich wie ein ROM einsetzen; sein Inhalt bleibt beim Betrieb unverändert. Er läßt sich aber durch Einsatz von UV-Licht löschen. Ein gelöschtes EPROM kann mit Hilfe eines "EPROM-Brenners", der an einem Rechner angeschlossen ist, neu programmiert werden. Anschließend schützt man sein "Löschfenster" mit einem Aufkleber gegen den Einfall von UV-Licht und somit gegen unbeabsichtigtes Löschen. EPROMS werden überall da eingesetzt, wo Anwender ohne industrielle Mittel Festprogramme verfügbar machen oder alte ROM-Versionen updaten wollen. EPROMS stellen im Lesezugriff die langsamste Art der Speicherbausteine dar.

A	F	A'	F'
B	C	B'	C'
D	E	D'	E'
H	L	H'	L'

Indexregister IX	
Indexregister IY	
Stackpointer SP	
Prg.-Counter PC	
Interrupt-register I	Refresh-register R

Der Registersatz der Z80-CPU. Eine besondere Rolle spielen dabei die Indexregister sowie "Stackpointer" (Stapelzeiger) und "Program-Counter" (Befehlszähler)

Register und Flags

Bevor nun die ersten Assembler-Zeilen codiert werden, gilt es, noch ein paar Vokabeln zu lernen. Register sind inzwischen als Schubladen für 8- und 16-Bit-Werte bekannt. Für unser Beispielprogramm machen wir erstmals Gebrauch davon. Allerdings sind wir genügsam und benutzen nur die Register "D", "E" und "IX", außerdem die als Doppelregister wirkenden "H" und "L".

Sehen wir uns nun das folgende kleine Assembler-Listing an:

```
150 'ORG &A000 ;      Startadresse festlegen
270 'LD L,(IX+2);     Adreß-Parameter über-
                      geben
280 'LD H,(IX+3);     und ins HL-Register
                      laden
290 'LD (HL),E ;      Low-Byte speichern
300 'INC HL ;         Adresse auf High-Byte
310 'LD (HL),D ;      High-Byte speichern
320 'RET ;            Zurück zum BASIC
```

Ein Blick auf die Details:

In Zeile 150 wird die Startadresse des Programms festgelegt. Bei &C000 bis &FFFF liegt weitgehend der Bildschirmspeicher. Das Betriebssystem befindet sich im Bereich von &0000 bis &3FFF. Es wird beim BASIC-Betrieb zwar weitgehend ausgeblendet, dennoch ist dieser Bereich keine sichere Bleibe für einen "blinden Passagier" in Maschinensprache. Eine gute Startadresse könnte aber auf den ersten Blick der darauf folgende Bereich sein. Er beginnt bei hexadezimal 4000, kurz 4000h oder in Assembler-Schreibweise "&4000".

Leider läßt jedoch BASIC keine Maschinenprogramme unterhalb der HIMEM-Grenze zu. Mit der Systemvariablen HIMEM wird ja bekanntlich die Obergrenze für BASIC-Programme im Speicher festgelegt. Will man ein Maschinenprogramm in den Speicher schmuggeln, setzt man zunächst HIMEM herunter. Wer also darauf bestünde, die DPOKE-Routine bei &4000 zu starten, müßte HIMEM auf &3FFF setzen:

```
100 HIGHMEM=&3FFF:REM das ist Quatsch
```

Jetzt wäre allerdings nur noch sehr wenig Platz fürs BASIC-Programm – genauer: nicht mehr als die 16 kByte, hinter denen sich das Betriebssystem versteckt. Es ist sehr viel besser, die Startadresse so hoch wie möglich anzusetzen – damit man möglichst viel Platz für weitere Programme übrigbehält. Im Beispiel haben wir für die Startadresse daher die ziemlich unkritische Stelle &A000 gewählt. Damit lassen wir noch rund 40 Kilobyte für BASIC-Programme frei und reservieren 8192 Bytes an "Maschinenraum".

Doch nun zurück zu den einzelnen Assemblerbefehlen der DPOKE-Routine:

LD ("Load") ist ein sogenannter Transportbefehl. Er transportiert 8-Bit-Werte zwischen beliebigen Speicherstellen und Registern. Dabei wird immer erst das Ziel und dann die Quelle angegeben. In Zeile 270 bringen wir also den Wert, auf den "IX+2" zeigt, in das "L"-Register.

"INC Register" tut nichts weiter, als den Wert des angegebenen Registers um Eins zu erhöhen. Hier die einzelnen Komponenten unseres Beispielprogramms noch einmal im Überblick:

- Register H,L,D und E: Über sie transportiert der LD-Befehl alle Werte, die im Programm benötigt werden. Teilweise werden dabei zwei Register als Doppelregister verwendet (Zeile 290).
- Flags: Das Zero-Flag wird automatisch gesetzt, wenn der Routine keine Parameter übergeben wurden.
- IX: Wichtig! Dieses Register enthält die *Adresse* des letzten Elements. Neben dem direkten Zugriff auf Werte kann man auch "Hausnummern" übergeben – Hinweise, wo entsprechende Werte abgeholt werden können. Soll von einem Register der Wert nicht direkt verwendet, sondern als Adresse betrachtet werden, so ist das Register in Klammern zu setzen.

Will man also nicht den Wert von "IX", sondern den Wert der Adresse, die in "IX" steht, benutzen, so schreibt man "...,(IX)". Dabei sind auch in die Klammer integrierte arithmetische Ausdrücke wie "(IX-2)" oder "(IX+4)" möglich. Hier wird zu dem Wert im Register IX ein Zahlenwert addiert beziehungsweise davon subtrahiert.

Ein derartiges Vorgehen wird als "Pointer-Arithmetik" bezeichnet.

"Pointer" bedeutet "Zeiger": Die Adresse, die in Klammern angegeben ist, enthält ja nicht den gesuchten Wert, sondern "zeigt" nur auf eine Adresse, die das tut.

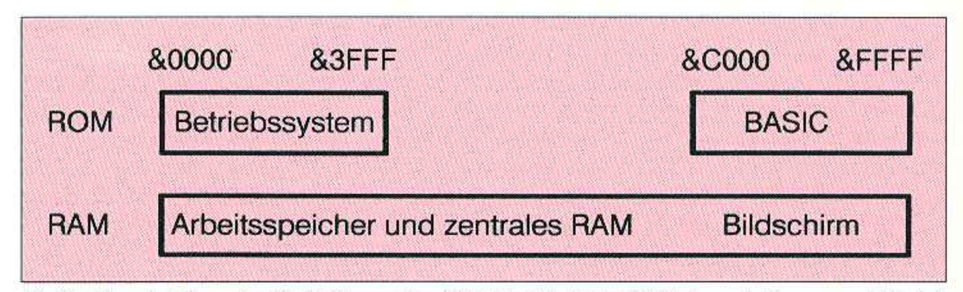
Start frei

Unsere DPOKE-Routine ist damit voll funktionsfähig. Nach dem Start des Programms über den Assembler ist sie im Speicher installiert. Speichert man die Routine als Binär-Datei über den Assembler ab, läßt sie sich wie folgt in jedes Programm einbinden und installieren:

```
...
110 ADRESSE=&A000
120 LOAD "DPOKE.BIN", ADRESSE
130 CALL ADRESSE
140 PRINT "DPOKE IST INSTALLIERT"
...
```

Der Aufruf erfolgt allerdings bislang noch nicht wie angestrebt über einen RSX-Befehl "|DPOKE", sondern über einen BASIC-"CALL"-Befehl: "CALL &A000, Adresse, Wert" würde zum gewünschten Ergebnis führen.

Um unsere Routine über "|DPOKE" aufrufen zu können, müssen wir das Betriebssystem des CPC zu Hilfe rufen. Glücklicherweise lauert dort ein Experte für das Einbinden von RSX-Befehlen.



Die Speichereinteilung der CPCs (den zweiten 64-kByte-Block des 6128 haben wir hier vernachlässigt)

Der Experte hat auch einen Namen und eine Hausnummer: Unter der Adresse "&BCD1" befindet sich die Betriebssystem-Routine "Logext", der wir nur noch einige Parameter übergeben.

Damit wir DPOKE erfolgreich einbinden können, müssen uns folgende Punkte klar sein:

- Der frei gewählte Name der Befehlserweiterung (DPOKE)
- Die Adresse, ab der das Programm abgelegt worden ist (&A000)
- Die Adresse eines 4 Byte großen Speicherraums, der zur Verwaltung des neuen RXS-Befehls gebraucht wird.

Die Adresse dieser vier Systembytes muß vor dem Aufruf von "Logext" ins HL-Register geladen werden. Außerdem ist das Registerpaar "BC" mit der Startadresse der Systemtabelle zu versehen. Diese Tabelle (TAB1) wiederum enthält als erstes die Adresse einer zweiten Tabelle. In dieser zweiten Tabelle (TAB2) steht der Name unseres neuen RSX-Befehls. Bei mehreren neuen Erweiterungen würden entsprechend weitere Namen folgen. Wichtig ist, daß in jedem Fall das höchstwertige Bit (Bit 7) des letzen Buchstabens eines Namens gesetzt wird. Beim "CPC-Assembler V 2.0" erledigt das automatisch die "DMX"-Anweisung. Bei anderen Assemblern muß man den letzten Buchstaben explizit definieren:

```
TAB2 DM "DPOK"
DB &C5          ;(oder: DB "E" + &80)
DB 0
```

DM definiert den ersten Teil des Namens. Der Pseudobefehl DB ("Define Byte") legt dann den Wert für das Byte fest, das den letzten Buchstaben bildet. Zum ASCII-Wert des betreffenden Buchstabens werden 128 (&80) addiert, es wird also das höchstwertige Bit gesetzt. Statt den gewünschten ASCII-Wert nachzuschlagen und dann die nicht sehr aussagekräftige Summe (hier &C5) zu setzen, darf man es sich bei den meisten Assemblern auch einfacher machen: die verständnisfreundlichere Variante haben wir oben in Klammern angegeben.

Wie geht es nun im Programm weiter? Auf die Adresse der Tabelle folgt der Sprungbefehl auf die eigentlichen Routinen der RSX-Befehle. Unser Beispielprogramm in der erweiterten Form sieht dann so aus, wie Sie es im Listing-Kasten im Anschluß an diesen Artikel finden.

Jetzt noch anschubsen

Nach dem Assemblieren des gesamten Programms muß die Routine einmal mit "CALL &A000" installiert werden. Danach verfügen Sie bis zum nächsten Warmstart (<Shift>-<Ctrl>-<Esc>) oder zum Ausschalten des Rechners über den "¦DPOKE"-Befehl.

Er kann wie jeder andere BASIC-Befehl im Direktmodus per Tastatur oder von einem Programm aus verwendet werden, läßt sich also auch in Listings einbauen. Die Syntax:

```
¦DPOKE, Adresse, Wert
```

Bekanntlich sind gerade bei etwas ungewohnten Themen schnelle Erfolgserlebnisse wichtig. RSX-Befehle bieten sich hier förmlich an, und so ganz nebenbei kommt man im Laufe der Zeit zu einem wirklich "persönlichen" BASIC.

Auf der nächsten Station unseres Assembler-Kurses wollen wir uns etwas mit den Flags, mit Schleifenprogrammierung ohne Knoten, außerdem mit Programmschutz und dessen Überwindung beschäftigen. Bis dahin "Gut Hack!".

Ulrich Schmitz/jg

Literatur:

1) Dullin, Retzlaff, Schneider, Strassenburg: CPC Tips & Tricks, Verlag Data Becker
2) Ulrich Schmitz: Ja und Nein, CPC Amstrad International 10/11'91, S. 32-34


```
100 '; Programm DPOKE.ASM
110 '; zum Assembler-Kurs, Folge 2
120 ';  (c) 1991 Ulrich Schmitz
130 '; & CPC International
140 ';
150 'ORG &A000         ;Startadresse festlegen
160 'LD BC,TAB1        ;Adresse der 1.Tabelle
170 'LD HL,SYSBYTE     ;Adresse  der 4 Bytes der RSX-
                        Verkettung
180 'CALL &BCD1        ;Routine "Logext" rufen
190 'RET               ;zurueck zu BASIC
200 'SYSBYTE DEFS 4 ;4 Bytes reservieren
```

```
210 'TAB1  DW TAB2    ;Link auf Tabelle 2
220 'JP DPOKE         ;Sprung auf Programm
230 'TAB2
240 'DMX "DPOKE"      ;Namenstabelle, 7.Bit gesetzt
250 'DB 0             ;Ende der Namenskennung
260 'DPOKE            ;- ab hier das Ur-Programm -
270 'LD L,(IX+2)      ;Adress-Parameter uebergeben
280 'LD H,(IX+3)      ;und ins HL-Register laden
290 'LD (HL),E        ;Low-Byte speichern
300 'INC HL           ;Adresse auf High-Byte
310 'LD (HL),D        ;High-Byte speichern
320 'RET              ;zurueck
330 'END
```

Jahresinhaltsverzeichnis 1991

SERVICE

Titel	Ausgabe	Seite
AMS-Line – Der heiße Draht zu AMSTRAD glüht wieder	4'5/91	35
AMS-Line – Tips und Tricks von AMSTRAD	6'7/91	79
AMS-Line – Der heiße Draht zu Amstrad	8'9/91	7
AMS-Line – Die Steckverbinder der CPC-plus-Reihe	10'11/91	31
AMS-Line – Der heiße Draht zu den Spezialisten	12'1/91/92	47

BERICHTE

Titel	Ausgabe	Seite
Nachhilfe für Profis – Was Tüftler alles zuwege bringen	2'3/91	113
Wie bei der Bundesbahn – Eine Modellbahnsteuerung für den CPC	4'5/91	36
Klein, kompakt, winzig – AMSTRAD's Neuvorstellungen auf der CeBIT	6'7/91	50
Ein Double für den CPC – Es gibt einen CPC-Kompatiblen	6'7/91	49
Ein Funkgehilfe namens CPC – Ein CPC in Wissenschaft und Hobby	8'9/91	8
Nachschub aus Ostwestfalen – CPC - Personality	8'9/91	11
Zwei Pocketcomputer – ATARI Portfolio und der SHARP IQ-8300M	8'9/91	14
Schachmatt in in zehn Spielen – Colossus-4 auf CPC und PCW	10'11/91	12
Das neue Kleid für CP/M – Betriebssystemaufsatz für CP/M	10'11/91	10
Großer Leserwettbewerb	10'11/91	13
Personality – Die Firma Wiedmann in Ismaning	10'11/91	14
"Englisches Wetter" für die CPCs – Die Business Computing'91 in London	12'1/91/92	19

PROGRAMME

Titel	Ausgabe	Seite
Zock around the Clock –Poker spielen auf dem CPC	2'3/91	8
Unterwegs im Kosmos – Sie sind Navigator des Raumschiffs ALTAIR	2'3/91	18
CRILION – Ein Spiel mit Ball und Steinen	2'3/91	27
Schluß mit dem Schreibkram – Füllen Sie Bankformulare auf dem CPC aus	2'3/91	32
Der Fehlerteufel hat zugeschlagen – Korrektur zum VOKABEL-TRAINER 12/1'91	2'3/91	38
Über alle Maße – Unterschiedliche Maßeinheiten	4'5/91	11
PIPELINE – Helfen Sie J.R. und Bobby beim Rohre legen	4'5/91	16
BONUSPROGRAMM – Die Beschreibung zum Bonusprogramm	4'5/91	28
Check your Papers – Verwaltung von Wertpapieren	4'5/91	30
Daten hin, Daten her – Eine Datenverwaltung für Ihren CPC	4'5/91	23
DIGIGLOBE – Die Welt auf dem Monitor	4'5/91	6
BONUSPROGRAMM: Einblicke – Das Monitorprogramm für Ihren CPC	6'7/91	13
Kontoverwaltung – Überwachen Sie Ihre Bankkonten	6'7/91	16
Ohne Moos nix los – Lohnsteuer-Berechnung	6'7/91	22
Dreh das Glücksrad – Ein Ratespiel für die ganze Familie	6'7/91	27
Belegverwaltung – Ordnung in den Fahrkostenquittungen	6'7/91	30
Mit Management geht's leichter – Ein Disketten-Manager für den CPC	6'7/91	37
CRASH – Spiel mit Bällen und Steinen	6'7/91	44
Herrscher über Land und Leute – Das Geschäftsleben des Mittelalters	6'7/91	6
Probleme mit der Energie – Energiekostenabrechnung mit dem CPC	8'9/91	16
Billy Bubble – Ein Geschicklichkeitsspiel	8'9/91	19
Aus klein mach groß – Schriften in beliebiger Größe auf dem CPC	8'9/91	25
Einem Oldie Dampf gemacht – CPC-Assembler neu aufgelegt	8'9/91	30
Ich wollt'ich wär'ein Wurm – In HighSpeed durch Labyrinth	8'9/91	37
Bonusprogramm: Herberts Little Helper – Heinzelmännchen haben's schwer	8'9/91	50
Der Faden der Ariadne – KI auf dem CPC	10'11/91	19
Gobang – Gobang zu Zweit	10'11/91	28
3D-Mersi – 3D-Tetris	10'11/91	22
RSX-Professional – große RSX-Bibliothek	12'1/91/92	20
Elektronikers Freund – Arbeit mit Widerständen erleichtert	12'1/91/92	
Frisch kopiert und editiert – Die Bonusprogramme	12'1/91/92	34

HARDWARE

Titel	Ausgabe	Seite
Port-Erweiterung für den CPC	2'3/91	40
AMSTRADs neuer 386er – Der PC3386 SX im Kurztest	2'3/91	42
The CPC Connection – Wie man zwei CPCs untereinander verbindet	4'5/91	52
Ein Nagetier für den CPC – GEOS-Maus am CPC	4'5/91	50
Der absolute Hammer – PC-Emulator auf dem CPC	4'5/91	48
Multifloppy – Alle Diskettenformate unter einem Hut	6'7/91	78
CPC Digiblaster – Musik aus dem Druckerport	8'9/91	48
Die Neuen sind da –Vorabinformationen zum neuen PCW	10'11/91	117

CP/M

Titel	Ausgabe	Seite
Statistik in Pascal – Statistik mit Turbo-Pascal	10'11/91	51
Dateien-Kompression – Der Huffmann-Algorithmus im Detail	10'11/91	42
Pascal-Kolleg – Den Logarithmen auf der Spur	10'11/91	48
PASCAL-Kolleg: Schöne Schleifchen – Schleifen in Turbo-Pascal und BASIC	12'1/91/92	35
Alles in Ordnung? Sortieralgorithmen	12'1/91/92	36
Viermal C – Vier C-Compiler im Test	12'1/91/92	40
Inline für Z80 – Austestung von Inline-Code in Turbo-Pascal	12'1/91/92	44

SOFTWARE-REVIEW

Titel	Ausgabe	Seite
Spiele, Spiele, Spiele – Spiele-Ecke für den CPC	6'7/91	70
Von der Antike ins 20. Jahrhundert – Infos zu aktuellen Spielen	8'9/91	71
Action, Simulation und mehr – Infos zu aktuellen Spielen	10'11/91	92
Heißer Stoff für kalte Nächte – Spielereview	12'1/91/92	10
Neues aus Frankreich – Brandneue Spiele aus Frankreich	12'1/91/92	17

KNOW-HOW

Titel	Ausgabe	Seite
Assembler-Ecke – Die Maschinensprache stellt sich vor	10'11/91	32
Weg mit dem GOTO – BASIC-Kurs	12'1/91/92	72

Titel	Ausgabe	Seite
Schwertgeschärfte Ohren – Der Weg zum eigenen Rollenspiel	12'1/91/92	76
Binärdateien trickreich verschlüsselt – Programme schützen mit der "Alkatrax"-Codierung	12'1/91/92	80
Assembler-Ecke: Gewitzt und fix mit RSX – Wie man dem BASIC neue Befehle unterjubelt	12'1/91/92	83

ANGELESEN

Titel	Ausgabe	Seite
Bücher – Drei interessante Bücher	10'11/91	41

TIPS & TRICKS

Titel	Ausgabe	Seite
Der 664/6128-Emulator – Emulator für den 464	2'3/91	44
CPC-Listings-Service – Listing-Abtippen leicht gemacht	2'3/91	46
100 DM für 1 kByte	2'3/91	48
Sprite-Characters – Eigene Zeichensätze	2'3/91	52
Stripp-Off – Textdateien universell machen	2'3/91	74
Gepatchtes Einerlei – Peeks und Pokes für CPC Programme	2'3/91	76
SUPERFORM v1.0 – 426 kByte auf CPC-Disketten	2'3/91	78
Testing, Testing – Ein Tracer	4'5/91	77
In alle Himmelsrichtungen – Eine Ergänzung zu ARTWORX	4'5/91	92
100 DM für 1 kByte – tolle 1-kByte-Programme	4'5/91	78
Am Character erkennt man den Druck – Selbstdefinierte Zeichensätze	4'5/91	84
Farbscreens auf Papier – Hardcopies auf Farbnadeldruckern	6'7/91	80
Pack your BASIC – Hilfsprogramme für BASIC	6'7/91	82
100 DM für 1kByte – Die nicht zu verachtenden Kleinigkeiten	6'7/91	86
Harte Sachen – Kleine Basteleien am CPC	8'9/91	51
MONA-MAXAM – MONA-Dateien unter MAXAM weiterverarbeiten	8'9/91	77
100 DM für 1kyte – und wieder schlagen wir zu	8'9/91	78
100 DM für 1 kByte – Sechs Spitzenprogramme	10'11/91	77
Floppy im Schnellgang – Lesen und Schreiben beschleunigt	10'11/91	80
Artworx-Druckerroutine – Artworx-Bilder in Spitzenqualität	10'11/91	87
Safer-Tape – Kassettenjustierung mit Software	10'11/91	81
Gamers-Message – Tips zu alten und neuen Spielen	10'11/91	83
100DM für 1 kByte – sechs der kurzen Superprogramme	12'1/91/92	48
Gamer's Message – "Erste Hilfe" für den Spielefreund	12'1/91/92	51
GIGACAD Grafikkonvertierung unter CP/M	12'1/91/92	70

PCW

Titel	Ausgabe	Seite
Termin'91 – Ein Terminkalender unter LocoFile2	2'3/91	79
Aller guten Dinge sind drei – MICRO Design II, PRO SCAN und Maus	2'3/91	80
Grafik ohne Ende – JOYCE, Vol. 3 mit Zusätzen	2'3/91	82
PROTECT – Datenschutz in dBASE II	2'3/91	87
Datenverluste Sicherer Arbeiten mit Laufwerk B	2'3/91	89
Wer LocoScript sagt... – Das Programm Super-Lindex von FSS	2'3/91	90
Auftrieb für den PCW – Neue Soft- und Hardware	2'3/91	92
LocoScript-Tips – Verbesserungen zu LocoScript	4'5/91	93
Dolmetscher II – CP/M wird eingedeutscht	4'5/91	100
HF-Tapete – Ein Hilfsmittel für jeden Elektroniker	4'5/91	96
DateSet – Datumseingabe mit Komfort	4'5/91	94
Ein exzellentes Trio – LocoScript, LocoFile und LocoMail	6'7/91	90
Update-Daten LocoScript PC – Neue Features	6'7/91	91
Funktion – Eine Funktionserweiterung für Turbo Pascal	6'7/91	92
ATOMIC – Strategiespiel mit Atomen	6'7/91	94
News, News, News – Brandheiße Infos zu JOYCE-Neuheiten	6'7/91	99
Im Herzen des Joyce – Der Speicheraufbau des JOYCE	6'7/91	100
Grafik ohne Ende – Programme mit DTP-Grfiken verfeinern	8'9/91	83
In Herzen des Joyce (2) – Der Kommunikationstreibers	8'9/91	89
Ein exzellentes Trio (2) – Eine Inventarliste erstellen	8'9/91	94
Software für den Super-Joyce – LocoScript und CP/M auf 720-kByte-Disk	8'9/91	96
Joyce Power – PCW beschleunigt	8'9/91	98
Im Herzen des Joyce (3) – Bildschirm und Zenith Z19Z29-Terminal-Emulation	10'11/91	100
Utilities sind überall – Viele nützliche Tips für den PCW	10'11/91	105
Schluß mit dem Schreibkram – Überweisungs- und Lastschriftdruck	10'11/91	106
Dreh'Dir einen – Glücksrad für den PCW	10'11/91	111
Im Herzen des Joyce: Der Zustand des Rechners – Wo kommen die Statusmeldungen her?	12'1/91/92	90
Ein Hauch von Lexus – Das programmierte Lexikon	12'1/91/92	96
BOUNCE – Bounce lädt zu einem flotten Spielchen ein	12'1/91/92	102
Grafiksystem selbstgemacht – Ein Grafiksystem zum Abtippen	12'1/91/92	108
Für Sie gespielt – Die Klassiker "Trivial Pursuit" und Scrabble	12'1/91/92	114
Locoscript, zum 2.3ten – Eine neue Version wird vorgestellt	12'1/91/92	116

PC

Titel	Ausgabe	Seite
Do you speak English ...? – Der FB-TRANSLATOR	2'3/91	93
Starkes Werkzeug in neuer Aufmachung – PC Tools 6.0, deutsch, im Kurztest	2'3/91	98
Der Kommandant und seine Crew – NC 3.0 und NORTON UTILITIES 5.0 D	2'3/91	100
Es geht auch anders... – Aktuelle Public Domain und Shareware	2'3/91	105
"Harte Copies", feiner Druck – Bildschirmausdrucke mit Pizazz Plus 2.0	2'3/91	109
COMMANDER KEEN – Ein Shareware-Spiel der Superlative	4'5/91	106
Aus allen Ländern – Neues aus Public Domain und Shareware	4'5/91	108
Weg mit den Karteikästen – Die Datenverwaltung ro-Kartei 1.0	4'5/91	112
Wörter, Wörter – Das PC-Wörterbuch Englisch	4'5/91	116
Des Bürokraten liebstes Kind – Ein Formularmanager	4'5/91	114
Die Maus am Stift – Eine interessante Maus-Variante	4'5/91	117
Schnittstelle zu Musik – Die SOUND-BLASTER-Karte	6'7/91	104
Neues aus aller Welt – PD und Shareware frisch auf den Tisch	6'7/91	110
Put'em in Your trouser pocket – Der CASIO SF 9500 Pocket-Computer	6'7/91	112
Auf ein neues – WORD 5.5 wird vorgestellt	8'9/91	106
Vergiß mein nicht – Der NVR beim PC 1512/1640	8'9/91	110
Alles unter einem Dach – Die Arbeitsoberfläche WORKS	8'9/91	112
Mausanpassung – Turbo 6 und Maus versöhnt	8'9/91	116

CPC

Canon S-300 am CPC

Meine Absicht ist es, die Schreibmaschine Canon S-300 mittels des Parallelinterfaces "Canon-30" als Typenraddrucker an meinen CPC anzuschließen. Können Sie mir mitteilen, wie das Textprogramm "Context CPC" an diese Konfiguration angepaßt wird?

Klaus Theulke
Halfing

Speziell für die Canon-Maschine haben wir leider keine angepaßte ConText-Version parat. Das bedeutet: Handarbeit. Alle Buchstaben außer den deutschen Umlauten, ß, Paragraph und noch ein paar Sonderzeichen müßten bei Verwendung von "Normalschrift" im Programm schon von Haus aus richtig gedruckt werden. Spannend wird es, wenn es um die oben erwähnten "Ausnahmen" sowie um Fettdruck, Hoch- und Tiefstellen geht. Dann gilt es nämlich, im Programmcode von ConText CPC die Steuersequenzen zu ändern.

Dazu laden Sie bei der kommerziellen ConText-Version die Datei "Context.Bas" und tragen zwischen den Zeilen 110 und 120 mit Hilfe von POKE-Befehlen die richtigen Steuercodes ein. An welche Adressen für welchen Zweck gePOKEt werden muß, steht im Handbuch zu ConText CPC.

Das veränderte Basic-Programm wird abgespeichert und sorgt dann nach dem Start für die richtige Druckeranpassung. Schwieriger sieht es da bei den Umlauten und Sonderzeichen aus – um sie zu verschieben, muß man die Zeichensatztabelle decodieren und ändern.

Wir wollen gern versuchen, die nötigen Schritte aufzudröseln und an einem Beispiel vorzuführen, bitten aber dafür um Geduld bis zum nächsten Heft. Für die POKEs brauchen Sie natürlich die Codes, die Ihre Canon-Typenradmaschine für Umlaute und Steuerbefehle benutzt. Diese müssen Sie der technischen Dokumentation Ihres Interfaces entnehmen (im Zweifel die Herstellerfirma fragen).

Vielleicht hat ja auch der eine oder andere unserer Leser schon eine fertige ConText-Anpassung für Typenradmaschinen auf Lager?

Red.

CPC

1-kByte-Programm "Clock"

In meinem 1-kByte-Programm "Clock" aus der CPC 8/9'91 hat sich leider ein kleiner Fehler eingeschlichen, wodurch der Stundenzeiger nicht so funktioniert, wie er soll.

Um den Fehler zu korrigieren, ändern Sie bitte folgende Zeilen:

```
10230 IF q<-270 THEN 10235
      ELSE 10250
10235 h=h-0.5
10240 MOVE x,y:DRAW
      x+(g/2.4)*COS(h+0.5),
      y+(g/2. 4)*SIN(h+0.5),0
10250 MOVE x,y:DRAW
x+(g/2.4)*COS(h),y+(g/2.4)*
SIN(h),13:RETURN
```

Nun funktioniert das Programm einwandfrei.

Thorsten Giertz
Remchingen

Leserbriefe 10/11'91

Fanzines

Im Brief des Herrn Maurice Homan aus Holland werden etliche deutsche Fanzines mit Namen benannt, jedoch keine Adressen angegeben. Da sicherlich einige Leser Interesse an einer dieser Mini-Zeitschriften haben, hier die Bezugsadressen, Preise und die Thematik/Lesergruppe:

- *Crackers International (Thriller & Weee)*
  Preis: 1,- DM + frankierter DIN-A5-Rückumschlag
  Redaktion: CI, PLK 063863 A, 6430 Bad Hersfeld
  Bezugsquelle:
  für Heft 7 siehe "Redaktion" ab Heft 8: MB-Versand, Postfach 501132, 5000 Köln 50

Wir sind ganz Ohr...

...für Ihre Fragen und Anregungen, was unser Heft betrifft – und alles andere rund um CPC oder PCW. Wollen Sie ein selbstgeschriebenes Programm im Heft unterbringen? Kennen Sie ein brandneues Programm, über das wir unbedingt etwas schreiben sollten? Kommen Sie mit einem der abgedruckten Listings nicht klar? Oder wollen Sie einfach mal mit einem von uns eine Runde fachsimpeln?

Dann ist unsere

Redaktions-Hotline

auch für Sie interessant.

Sie sind herzlich eingeladen – wir wissen zwar nicht auf jede Frage die richtige Antwort, aber mancher Knoten löst sich trotzdem schon am Telefon.

Jeden Dienstag von 16 bis 19 Uhr

hängt das komplette Redaktionsteam der CPC Amstrad International für Sie an der Leitung. Hier unsere Telefonnummern:



Jörg Gurowski (jg)
0 56 51 / 8 09 - 7 51



Ralf Schößler (rs)
0 56 51 / 8 09 - 7 52



Peter Schmitz (sz)
0 56 51 / 8 09 - 7 53

Hallo, Clubs!

Wir möchten mal wieder etwas für eine lebendige Clubszene unter den deutschen CPC- und PCW-Freunden tun. Daher richten wir ab sofort eine **Clubseite** in der CPC Amstrad International ein. Diese Seite ist dann Eure Seite! Ihr könnt darauf zum Beispiel:

- Eure Club-Adresse veröffentlichen und Mitglieder werben
- auf Veranstaltungen hinweisen oder über bereits Gewesenes berichten
- die Nummern von interessanten Mailboxen oder Btx-Seiten austauschen
- eine Selbstdarstellung Eures Clubs bringen, natürlich auch mit Fotos (bitte nur schwarzweiß!)

Klar, daß die Veröffentlichung auf der Clubseite für Euch völlig kostenlos ist. Ebenfalls klar, daß Ihr so auf einen Schlag den größten Teil aller User erreicht, die es noch gibt. Über zehntausend Leute kaufen die CPC Amstrad International alle zwei Monate – macht Euch das zunutze.

Achtet bitte darauf, daß wir produktionsbedingte Vorlaufzeiten haben. Wenn Ihr also beispielsweise auf einen Flohmarkt im Februar hinweisen wollt, muß die entsprechende Meldung bis Weihnachten bei uns sein. Dann bekommen wir sie noch ins Heft 2/3'92 hinein, das im Januar erscheint. Schreibt an:

DMV – Daten- und Medienverlag
Redaktion CPC Amstrad International
– Clubseite –
Postfach 250
3440 Eschwege

Inhalt: Szenen-News, aktuelle Spiele- und Demotests, Erwischt-Stories...
Erscheinungsweise: unregelmäßig

- *CPC Power International (Mike Behrendt)*
Preis: 5,-DM inkl. Porto
Redaktion/Vertrieb: CPC-Power, Postfach 501132, 5000 Köln 50
Inhalt: Anwenderkurse (Basic, Assembler, Forth...), Tests von Spielen und Anwendungen, Cover Tape, Berichte...
Erscheinungsweise: alle zwei Monate

- *Crackers International (KNS)*
Preis: 2,– bis 3,– DM
Adresse: CI, PF 117, 5412 Ransbach-Baumb.
Zu Inhalt und Zielrichtung habe ich leider keine näheren Angaben.

- *CPC Challenge (OAS)*
Preis: 3,-DM inkl. Porto
Redaktion: PLK 124 105 C, 2300 Kiel
Vertrieb: Postfach 501132, 5000 Köln 50
Inhalt: Spieletests, Pokes, Szeneberichte, Party-Stories

- *Frame Fly (Pluton of Commotion)*
Diskettenmagazin, erscheint je nach Bedarf und Zeit

Die Bezugsadresse hierfür liegt mir leider nicht vor.

Mike Behrend
Köln

CPC

Argumentenübergabe in Assembler

Ich beschäftige mich seit einiger Zeit mit Assemblerprogrammierung. Nun möchte ich natürlich auch die Floppy-Befehle verwenden. Ich habe aber Probleme mit der Argumentenübergabe. Können Sie mir einen Tip geben, wie das funktioniert? Selbst langwieriges Probieren hat bislang zu keinem Erfolg geführt.

Guthard Manfred
Bad Königshofen

Die Argumentenübergabe an einen RSX-Befehl geschieht so, wie es das Listing im Kasten am Beispiel von |REN,"TEST.NEW","TEST.OLD" in Assembler zeigt.
Dies ist natürlich nur eine Art "Schnellimbiß". Wer mehr "Futter" braucht, sollte unseren Assemblerkurs verfolgen!

Red.

CP/M

Small-C-Compiler

In der letzten Ausgabe der CPC International (10/11'91) las ich voller Interesse den Artikel über Datenkompression und war dann sehr enttäuscht, daß das abgedruckte Beispielprogramm in der Programmiersprache C geschrieben wurde.

In Ihrem Artikel schreiben Sie, daß der verwendete Compiler über Ihren Verlag zu beziehen ist.

Ich würde nun gerne wissen, was im Lieferumfang von Small-C enthalten ist und was es kostet.

Stefan Heier
Iffezheim

Leider ist es uns nicht gelungen, den Small-C-Compiler in unser Programm aufzunehmen.

Sie können ihn jedoch bei der Firma Weeske, Backnang, für rund 20 DM bekommen. Im Lieferumfang befindet sich eine 3-Zoll-Diskette sowie ein deutsches Anleitungsheft, das unter anderem eine Erklärung der Unterschiede zwischen Small-C und dem Standard-C bietet. Ebenfalls enthalten ist ein Editor für die Quelltexte.

Red.

```
       LD HL,BEF
       CALL
       #0BCD4
       RET NC
       LD A,2 ;An-
       zahl der
       Argumente
       LD IX,ARG
       CALL #01B
       RET
BEF:   DEFM 'RE'
       DEFB
       'N'+#80
ARG:   DEFW ARG1   ;1. Argu-
                   ment
       DEFW ARG2   ;2.Argume-
                   nt
ARG1:  DEFB 8      ;Der Name
                   ist 8
       DEFW: NAME  ;Zeichen
                   lang
       ;und be-
       ginnt
       ;bei
       Adresse
       ;"NAME"
       NAME:       DEFM
                   'TEST.OLD'
ARG2:  DEFB 8      ;Der Name
                   ist 8
       DEFW:
       NAME1       ;Zeichen
                   lang
       ;und be-
       ginnt
       ;bei
       Adresse
       ;"NAME1"
       NAME1:      DEFM
                   'TEST.NEW'
```

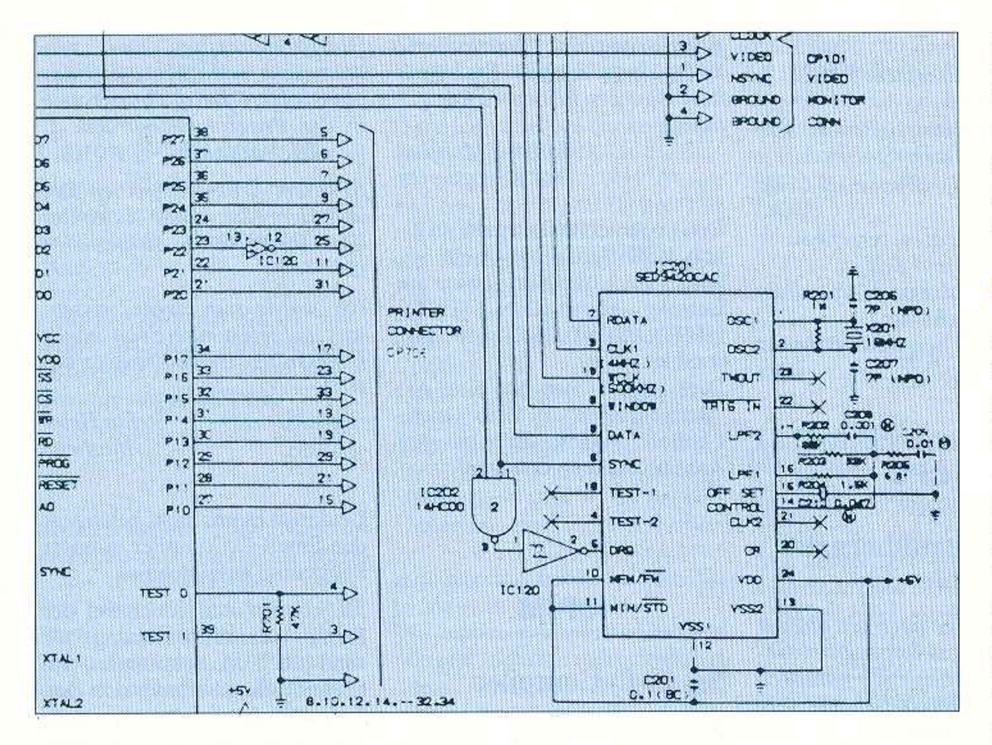
In eigener Sache: "'Tschuldigung!"

Einige von Ihnen, liebe Leser, dürften mit Recht ein wenig verärgert sein. Grund: Mancher Leserbrief, der bereits vor längerer Zeit an uns geschickt worden ist, wartet bis heute noch auf seine Beantwortung.
Tatsächlich haben wir gewisse Schwierigkeiten, mit der Menge an Leserzuschriften fertigzuwerden. Immerhin erreichen uns ja nicht bloß Briefe, die das Heft betreffen, sondern auch Fragen und Vorschläge zu Softwareprodukten, die vom DMV vertrieben werden. Außerdem gibt es noch jede Menge Briefe, in denen Hilfe bei Hard- oder Software-Anpassungsproblemen gesucht wird. Mit der Beantwortung all dieser Fragen ist ein riesiger Recherche-Aufwand verbunden, dem unsere kleine Redaktion einfach nicht gewachsen ist. Immerhin müssen wir ja zu dritt (zwei Redakteure, ein Chefred) termingerecht Themen fürs Heft erarbeiten, Software testen, Artikel redigieren, Meldungen schreiben, für Bildschirmfotos sorgen, Beiträge auf Zuverlässigkeit überprüfen, und, und, und... – kurz, alles tun, damit Sie alle zwei Monate wieder eine CPC International in den Händen halten können, die ihr Geld wert ist. Außerdem machen wir die DATABOX-Disketten – viel Schweiß und Erleichterung, wenn wieder eine Ausgabe zusammengestellt, getestet und zum Kopieren fertig ist.
Bei den Leserbriefen gilt es natürlich zuerst, die interessantesten Anliegen fürs Heft aufzubereiten. Erst dann kommen die "Einzelanliegen" zu ihrem Recht. Wir können also nur herzlich um Entschuldigung bitten, wenn über all der Termin-Hektik, dem Umzug und dem Mannschaftswechsel der letzten Monate einige Fragen offengeblieben sind. Um in Zukunft schneller reagieren zu können, lassen wir hiermit einen

Hilferuf

los: Wir suchen freie Mitarbeiter, die uns auf Honorarbasis helfen, Leserfragen zu Hard- und Software von CPC oder PCW zu beantworten. So könnte sich etwa ein passionierter Bastler entsprechend seinem Interessenschwerpunkt um alles kümmern, was Schnittstellen, Umbau und Erweiterung des Computers betrifft. In puncto Honorar zeigen wir uns durchaus großzügig, und besondere sprachliche Qualitäten sind nicht erforderlich: Wenn's um die Veröffentlichung im Heft geht, polieren wir die Antworten gern ein bißchen journalistisch auf.
Na, wie wär's? Hier sind besonders die "Spezis" von Fanzines und Clubs gefragt: Wollt Ihr uns helfen, besser zu werden? Ein Versuch lohnt sich sicher auch für Euch. Schreibt an:

DMV – Daten- und Medienverlag
Redaktion CPC Amstrad International
Stichwort: freie Mitarbeit
Postfach 250
3440 Eschwege



Im Herzen des JOYCE

Teil 4: Der Zustand des Rechners

Wenn wir über den Zustand unseres Computers sprechen, hat dies natürlich weniger mit dem materiellen Zustand, sprich dem 'Alter' des Rechners zu tun, sondern vielmehr mit dem operationellen Zustand, auch als 'STATUS' bezeichnet. Dem Nutzer wird dieser Status über sogenannte Statusmeldungen mitgeteilt, damit er über eine vorhandene Fehlerquelle oder die eventuellen 'Unzulänglichkeiten' seines Rechners informiert ist. Wichtige Änderungen im Status werden meist automatisch angezeigt (zum Beispiel Fehler), andere Informationen können über die Tastatur oder aus Programmen heraus abgerufen werden.

Im heutigen Teil unseres Streifzugs durch die BIOS-Tiefen der PCW-Rechner befassen wir uns mit den Möglichkeiten des JOYCE, uns seinen 'Status' mitzuteilen. Diese Mitteilungen erfolgen entweder über die Statuszeile (die untere Bildschirmzeile) oder direkt auf dem Bildschirm, wenn die Statuszeile ausgeschaltet ist. Der Terminal-Emulator, der die Kommunikation mit dem Anwender-Programm steuert, hält auch hierfür die entsprechenden Routinen bereit. Bevor wir uns jedoch diesen Routinen zuwenden, benötigen wir noch einige Kenntnisse über den Aufbau solcher Meldungen.

Jede Statusmeldung setzt sich aus mehreren Textfragmenten zusammen, die ebenso wie die Statusmeldungen selbst, durch eine entsprechende Text-Nummer gekennzeichnet sind.

Erst beim Aufruf werden diese Textfragmente, zusammen mit eventuell benötigten Kontrollinformationen, Zahlenwerten und so weiter, zu einer verständlichen Statusmeldung zusammengefügt.

Von den insgesammt 70 Text-Nummern werden nur 25 (siehe Tabelle 1) für den direkten Aufruf der Statusmeldungen benutzt, weitere 45 definieren die Textfragmente, die zur Erzeugung dieser Meldungen benötigt werden.

Ein Beispiel:

Die Statusmeldung mit der Aufrufnummer 0 (siehe Tabelle 1) ist in der Tabelle der Statusmeldungen ($1920 - $1C1F in Speicherbank #0) wie folgt abgespeichert:

`FE C3 BC C1 FF` Für die komplette Meldung werden also nur fünf Bytes benötigt, die beim Aufruf die jeweils benötigten Textfragmente zur eigentlichen Meldung zusammenfügen. Durch die Verwendung von Steuercodes innerhalb dieser Textfragmente ist es möglich, Zahlen, Gerätenamen, etcetera ebenfalls in die Statusmeldung einzufügen.

Gesteuerte Status-Ausgabe

Mit entsprechenden Kenntnissen über Bedeutung und Verwendung dieser Steuerzeichen lassen sich somit ohne Schwierigkeiten Tabellen zur Ausgabe 'eigener' Statusmeldungen aufbauen. Dazu muß der Adresseintrag der BIOS-Statustabelle (in Adresse $09CA) auf die eigene Tabelle umgeleitet werden. Dies setzt jedoch voraus, daß die eigene Statustabelle in der sogenannten COMMON-Memory (Adressbereich $C000-$FFFF) liegt, damit sie auch von den Systemroutinen erreicht werden kann.

Tabelle 2 faßt die benutzbaren Steuerzeichen zusammen, hier eine kurze Erläuterung:

Das Zeichen $F3 bedeutet, daß das nächste Ausgabe-Zeichen für die Meldung einen ASCII-Wert >127 hat und als solches ausgegeben werden soll. Normalerweise werden alle Zeichen >127 als Steuerzeichen betrachtet, mit $F3 wird es möglich, auch Sonderzeichen/Grafikzeichen in die Statusmeldung einzufügen. Beispiel: F3 A4 fügt das Copyright- Zeichen (ASCII-Code $A4) in die Statusmeldung ein.

Das Zeichen $F4 fügt Leerzeichen in die Ausgabe-Meldung ein. Die Anzahl der Leerstellen wird mit dem nächsten Zeichen festgelegt. Beispiel: F4 08 fügt acht Leerzeichen in die Statusmeldung ein. $F5 ermöglicht die Auswahl von drei verschiedenen Textfragmenten aus der Statustabelle in Abhängigkeit von den Bits 0 & 1 des D-Registers.

Beispiel: F5 12 43 25 19

Abhängig vom Inhalt des D-Registers (b1/b0) werden folgende Textfragmente in die Statusmeldung eingefügt:

D(b1/b0) = 00: Text-Nummer 18 ($12)

D(b1/b0) = 01: Text-Nummer 67 ($43)

D(b1/b0) = 10: Text-Nummer 37 ($25)

Das D-Register wird bei jeder Prüfung um zwei Bits nach rechts rotiert, so daß maximal vier Auswahlfelder zur Verfügung stehen.

Nach der Ausgabe des Textfragments werden automatisch die nächsten drei, dem Steuerzeichen folgenden Text-

Nummern übersprungen und die Ausgabe mit dem Fragment #25 ($19) fortgesetzt. Hieraus folgt, daß die Bit-Folge elf im D-Register für diese Ausgabeform nicht zulässig ist, denn sonst würde Meldung #25 doppelt ausgegeben.

Die Zeichen $F6 und $F7 ermöglichen ein Umschalten des Ausgabeattributes auf farbinvertierte Ausgabe ($F7) und normale Ausgabe ($F6).

Für die Umschaltung auf invertierte Ausgabe muß zusätzlich das Bit sieben des C-Registers gesetzt sein. Der Inhalt des C-Registers wird bei jeder Prüfung um eine Position nach links rotiert, so daß es möglich ist, aus bis zu acht Menue-Feldern ein einzelnes Feld invers zu setzen (wie zum Beispiel beim Drucker-Status).

Beispiel: F7 41 F6 gibt den Buchstaben 'A' innerhalb der Statusmeldung invertiert aus, wenn das MSB des C-Registers auf '1' gesetzt ist.

Das Steuerzeichen $F8 wandelt die im Register HL spezifizierte Binärzahl in eine Dezimalzahl um und fügt sie formatiert, das bedeutet mit vorangestellten Nullen, in die Statusmeldung ein. Hier sind nur dreistellige Zahlenwerte zulässig (000..999).

Beispiel: HL = $0014 , das Zeichen $F8 würde diese Zahl in 020 umwandeln und ausgeben.

Um einen beliebigen Text in die Statusmeldung einzufügen, benutzt man das Steuerzeichen $F9. Es werden jeweils sechs Zeichen eingefügt, die über das HL-Register addressiert werden. Leerstellen, die im Textstring enthalten sind, werden nicht mit ausgegeben. Beim JOYCE wird dieses Steuerzeichen für die Ausgabe der Gerätenamen (CEN, LPT SIO, ...) benutzt.

$FA, $FB und $FC werden wieder zur Zahlenumwandlung benutzt, wobei hier die auszugebende Zahl im DE-Register übergeben wird (siehe Tabelle 2). Die Ausgabe erfolgt unformatiert, also ohne vorangestellte Nullen.

Beispiel: FB wandelt die im E-Register enthaltene Binärzahl in eine Dezimalzahl um und fügt sie in die Statusmeldung ein.

Die Zeichen $FD und $FE geben einen Laufwerks-Bezeichner (A:, B:, ...) aus. Der Unterschied zwischen beiden Zeichen besteht darin, daß $FE nur dann eine Ausgabe veranlaßt, wenn mehr als ein Laufwerk vorhanden ist, $FD gibt das Laufwerk unabhängig von der Anzahl der vorhandenen Laufwerke aus.

$FF wird dazu benutzt, das Ende des Textfragments/der Statusmeldung zu kennzeichnen. Alle weiteren Zeichen, die einen ASCII-Wert >127 aufweisen, werden als neue Text-Nummer interpretiert, die eventuell in eine Statusmeldung oder ein anderes Textfragment eingefügt werden soll. Dazu wird einfach das MSB gelöscht, und man erhält die neue Text-Nummer. Auf diese Weise lassen sich eine große Anzahl verschiedener Meldungen bei relativ geringem Speicherplatz-Bedarf zusammenfassen. Dekodieren wir nun die fünf Bytes der oben angegebenen Statusmeldung #0, so erhalten wir folgendes Ergebnis:

$FE gibt den Lauferks-Bezeichner des aktuellen Laufwerks aus, wenn mehrere Laufwerke vorhanden sind, zum Beispiel:

```
A:
```

$C3 fügt das Textfragment #67 ($43) in die Meldung ein:

```
Laufwerk
```

$BC fügt das Textfragment #60 ($3C) in die Meldung ein:

```
nicht bereit
```

$C1 fügt das Textfragment #65 ($41) in die Meldung ein:

```
- Wiederholen, Ignorieren oder Abbrechen?
```

$FF bezeichnet das Ende der Statusmeldung und beendet die Ausgabe.

Routinen zur Bearbeitung

Damit Sie diese Informationen auch verwerten und 'eigene' Statustexte erzeugen können, folgt hier eine Beschreibung der vom Terminal-Emulator für die Bearbeitung von Statusmeldungen angebotenen System-Routinen (vergleich Tabelle 3):

TE STL INIT $0ABC
wird während der System-Initialisierung aufgerufen und setzt das Statuszeilen-Flag auf $FF (=ON).

TE STL OFF (ESC 0) $0AC2
löscht die Statuszeile und setzt das Statuszeilen-Flag auf 0 (=OFF).

TE STL ON (ESC 1) $0AC5
löscht die Statuszeile und setzt das Statuszeilen-Flag auf $FF (=ON).

TE STL PRINT MSG $0AD2
wird vom BIOS zur Ausgabe von Statusmeldungen benutzt und gibt den Status entweder auf der Statuszeile (wenn eingeschaltet) oder direkt am Bildschirm aus.

Statusausgabe beenden $0AF2

Für die Umschaltung auf invertierte Ausgabe muß zusätzlich das Bit sieben des C-Registers gesetzt sein. Der Inhalt des C-Registers wird bei jeder Prüfung um eine Position nach links rotiert, so daß es möglich ist, aus bis zu acht Menue-Feldern ein einzelnes Feld invers zu setzen (wie zum Beispiel beim Drucker-Status).

Beispiel: F7 41 F6 gibt den Buchstaben 'A' innerhalb der Statusmeldung invertiert aus, wenn das MSB des C-Registers auf '1' gesetzt ist.

Das Steuerzeichen $F8 wandelt die im Register HL spezifizierte Binärzahl in eine Dezimalzahl um und fügt sie formatiert, das bedeutet mit vorangestellten Nullen, in die Statusmeldung ein. Hier sind nur dreistellige Zahlenwerte zulässig (000..999).

Beispiel: HL = $0014 , das Zeichen $F8 würde diese Zahl in 020 umwandeln und ausgeben.

Um einen beliebigen Text in die Statusmeldung einzufügen, benutzt man das Steuerzeichen $F9. Es werden jeweils sechs Zeichen eingefügt, die über das HL-Register addressiert werden. Leerstellen, die im Textstring enthalten sind, werden nicht mit ausgegeben. Beim JOYCE wird dieses Steuerzeichen für die Ausgabe der Gerätenamen (CEN, LPT SIO, ...) benutzt.

$FA, $FB und $FC werden wieder zur Zahlenumwandlung benutzt, wobei hier die auszugebende Zahl im DE-Register übergeben wird (siehe Tabelle 2). Die Ausgabe erfolgt unformatiert, also ohne vorangestellte Nullen.

Beispiel: FB wandelt die im E-Register enthaltene Binärzahl in eine Dezimalzahl um und fügt sie in die Statusmeldung ein.

Die Zeichen $FD und $FE geben einen Laufwerks-Bezeichner (A:, B:, ...) aus. Der Unterschied zwischen beiden Zeichen besteht darin, daß $FE nur dann eine Ausgabe veranlaßt, wenn mehr als ein Laufwerk vorhanden ist, $FD gibt das Laufwerk unabhängig von der Anzahl der vorhandenen Laufwerke aus.

$FF wird dazu benutzt, das Ende des Textfragments/der Statusmeldung zu kennzeichnen.

Alle weiteren Zeichen, die einen ASCII-Wert >127 aufweisen, werden als neue Text-Nummer interpretiert, die eventuell in eine Statusmeldung oder ein anderes Textfragment eingefügt werden soll. Dazu wird einfach das MSB gelöscht, und man erhält die neue Text-Nummer. Auf diese Weise lassen sich eine große Anzahl verschiedener Meldungen bei relativ geringem Speicherplatz-Bedarf zusammenfassen. Dekodieren wir nun die fünf Bytes der oben angegebenen Statusmeldung #0, so erhalten wir folgendes Ergebnis:
$FE gibt den Lauferks-Bezeichner des

– Laufwerks-Status:

0 LW: Laufwerk nicht bereit - Wiederholen, Ignorieren oder Abbrechen?
1 LW: Diskette schreibgeschützt - Wiederholen, Ignorieren oder Abbrechen?
2 LW: Spur n Suche ohne Erfolg - Wiederholen, Ignorieren oder Abbrechen?
3 LW: Spur n, Sektor n Datenfehler - Wiederholen, Ignorieren oder Abbrechen?
4 LW: Spur n, Sektor n keine Daten - Wiederholen, Ignorieren oder Abbrechen?
5 LW: Spur n, Sektor n Adresseintrag fehlt -Wiederholen, Ignorieren oder Abbrechen?
6 LW: ungültiges Format - Wiederholen, Ignorieren oder Abbrechen?
7 LW: unbekannter Fehler - Wiederholen, Ignorieren oder Abbrechen?
8 LW: Diskette wurde gewechselt - Wiederholen, Ignorieren oder Abbrechen?
9 LW: falsche Diskette - Wiederholen, Ignorieren oder Abbrechen?

Die Angabe des Laufwerks (LW:) erfolgt nur, wenn zwei oder mehr Laufwerke vorhanden sind)

– System-Status:

10 CP/M Plus Amstrad Consumer Electronics plc v 1.4, 61K TPA, (zusätzliche Info ist abhängig von der vorhandenen Hardware)
16 Bitte Diskette für LW: einlegen, dann irgendeine Taste drücken (Meldung erscheint, wenn nur ein LW vorhanden ist und ein zweites LW vom System 'emuliert' wird)
17 GERÄT: nicht bereit - Wiederholen, Ignorieren oder Abbrechen?
19 Laufwerk ist LW: (Meldung erscheint, wenn nur 1 LW vorhanden und die Statuszeile eingeschaltet ist)

– Drucker-Status:

20 Drucker: Aktiv
21 Drucker: On Line
22 Drucker: Off Line
23 Drucker: Papier zu Ende
24 Drucker: Papier einlegen
25 Drucker: Andruck offen
26 Drucker: FEHLER - Unterlauf
27 Drucker: FEHLER - Drucker RAM
28 Drucker: FEHLER - Unzul. Befehl
29 Drucker: FEHLER - Druck-Fehler
30 Drucker: kein Drucker

Die Statuszeile für Drucker-Meldungen wird durch die folgenden Meldungen erweitert, die beim Aufruf über das D-Register definiert werden:
- in Zeile n / Seitenanfang
- LF
- FF
- Entwurf / Korrespondenz
- PE ignor:Aus / PE ignor:Ein
- Hex:Aus / Hex:Ein
- Reset

Code:	Bedeutung:
$F3	der ASCII-Code des nächsten Ausgabe-Zeichens ist >127 ($7F)
$F4	zusätzliche Leerzeichen ausgeben
$F5	fügt Text aus der Statustabelle in die Statusmeldung ein (Auswahl durch D (b0/b1), siehe Text)
$F6	Ausgabe-Attribute auf 'normal' setzen
$F7	Ausgabe-Attribute auf 'invers' setzen (C b7 muß '1' sein, siehe Text)
$F8	formatierte Ausgabe einer Dezimalzahl (000..999, HL=Zahl)
$F9	Text einfügen (ohne Leerstellen, z.B. Gerätenamen)
$FA	unformatierte Ausgabe einer Dezimalzahl (0..65535, DE=Zahl)
$FB	unformatierte Ausgabe einer Dezimalzahl (0..255, E=Zahl)
$FC	unformatierte Ausgabe einer Dezimalzahl (0..255, D=Zahl)
$FD	Laufwerks-Bezeichner (LW:) ausgeben
$FE	Laufwerks-Bezeichner (LW:) ausgeben (nur bei 2 und mehr LW)
$FF	Text-Ende (EOT)

Alle weiteren Codes mit ASCII-Werten >127 ($7F) werden als Text-Nummer interpretiert indem das MSB (Bit 7) gelöscht wird.

Tabelle 2: Steuercodes für die Erzeugung von Statusmeldungen

Funktion	Adresse / Parameter
TE STL INIT Eing.: Ausg.: Benutzte Register: A	Adresse: $0ABC keine Parameter keine Parameter
TE STL OFF (ESC 0) Eing.: Ausg.: Benutzte Register: A, BC, DE, HL	Adresse: $0AC2 keine Parameter keine Parameter
TE STL ON (ESC 1) Eing.: Ausg.: Benutzte Register: A, BC, DE, HL	Adresse: $0AC5 keine Parameter keine Parameter
TE STL PRINT MSG Eing.: Ausg.: Benutzte Register: A, BC, DE, HL	Adresse: $0AD2 B = Text-Nummer keine Parameter
Statusausgabe beenden Eing.: Ausg.: Benutzte Register: A, BC, DE, HL	Adresse: $0AF2 keine Parameter keine Parameter
TE SHOW DRIVE Eing.: Ausg.: Benutzte Register: A, BC, DE, HL	Adresse: $0AFD keine Parameter keine Parameter
Statuszeile Löschen Eing.: Ausg.: Benutzte Register: A, BC, DE, HL	Adresse: $0B17 keine Parameter keine Parameter
TE STL SET DRIVENUM Eing.: Ausg.: Benutzte Register: A, C	Adresse: $0B20 C = Laufwerksnummer (A=0, B=1, ...) keine Parameter
TE STL ASK Eing.: Ausg.: Benutzte Register: A	Adresse: $0B25 keine Parameter A = Statusz.-Flag ($00=OFF/$FF=ON)
TE STL PREPARE Eing.: Ausg.: Benutzte Register: A, BC, DE, HL	Adresse: $0B2A A = linke Spalte für die Text-Ausgabe B = Textnummer Alle anderen Register entsprechend der Ausgabemeldung keine Parameter
TE STL CLEAR Eing.: Ausg.: Benutzte Register: A, BC, DE, HL	Adresse: $0B4D keine Parameter keine Parameter
TE PTR STL PREPARE Eing.: Ausg.: Benutzte Register: A, BC, DE, HL	Adresse: $0B6B keine Parameter D = akt. Attribute Code E = Cursor-Ticker
TE PTR MSG PREPARE Eing.: Ausg.: Benutzte Register: A, BC, DE, HL	Adresse: $0B7D A = Textnummer BC & DE enthalten Status-masken entspr. der gew. Ausgabemeldung keine Parameter
TE PTR STL RESTORE Eing.: Ausg.: Benutzte Register: A, BC, DE, HL	Adresse: $0BA1 D = Attribute-Code E = Cursor-Ticker keine Parameter

aktuellen Laufwerks aus, wenn mehrere Laufwerke vorhanden sind, zum Beispiel:

`A:`

$C3 fügt das Textfragment #67 ($43) in die Meldung ein:

`Laufwerk`

$BC fügt das Textfragment #60 ($3C) in die Meldung ein:

`nicht bereit`

$C1 fügt das Textfragment #65 ($41) in die Meldung ein:

`- Wiederholen, Ignorieren oder Abbrechen?`

$FF bezeichnet das Ende der Statusmeldung und beendet die Ausgabe.

Routinen zur Bearbeitung

Damit Sie diese Informationen auch verwerten und 'eigene' Statustexte erzeugen können, folgt hier eine Beschreibung der vom Terminal-Emulator für die Bearbeitung von Statusmeldungen angebotenen System-Routinen (vergleich Tabelle 3):

TE STL INIT $0ABC
wird während der System-Initialisierung aufgerufen und setzt das Statuszeilen-Flag auf $FF (=ON).

TE STL OFF (ESC 0) $0AC2
löscht die Statuszeile und setzt das Statuszeilen-Flag auf 0 (=OFF).

TE STL ON (ESC 1) $0AC5
löscht die Statuszeile und setzt das Statuszeilen-Flag auf $FF (=ON).

TE STL PRINT MSG $0AD2
wird vom BIOS zur Ausgabe von Statusmeldungen benutzt und gibt den Status entweder auf der Statuszeile (wenn eingeschaltet) oder direkt am Bildschirm aus.

Statusausgabe beenden $0AF2
schließt die BIOS-Statusausgabe ab und schickt ein CR/LF an den Bildschirm, wenn die Ausgabe nicht auf der Statuszeile erfolgte.

TE SHOW DRIVE $0AFD
zeigt an, welches Laufwerk vom System unterstützt wird (A: oder B:), wenn nur ein Laufwerk vorhanden ist. Um eine Anzeige zu erhalten, muß die Statuszeile eingeschaltet sein.

Statuszeile löschen $0B17
löscht die Statuszeile, wenn sie eingeschaltet ist.

TE STL SET DRIVNUM $0B20
legt die Laufwerksnummer für TE SHOW DRIVE fest.

TE STL ASK $0B25
gibt das Statuszeilen-Flag zurück ($00/$FF).

TE STL PREPARE $0B2A
Vorbereitung und Aufruf der Statusausgabe auf der Statuszeile.

TE STL CLEAR $0B4D
löscht die Statuszeile (keine Prüfung, ob die Statuszeile eingeschaltet ist).

Die Routinen für die Ausgabe des Drucker-Status unterscheiden sich etwas von denen, die vom BIOS benutzt werden. Wenn das BIOS eine Statusmeldung ausgibt, wird die Statuszeile einfach überschrieben, der Hintergrund ist verloren. Bei der Ausgabe der Statusmeldungen für den Drucker wird der Hintergrund gesichert und nach Abschluß der Statusbearbeitung wiederhergestellt, es gehen also keine Bildschirm-Informationen verloren.

Will man diese Routinen für die eigene Status-Ausgabe benutzen, benötigt man jedoch zusätzliche, detailierte Kenntnisse der Interrupt-Programmierung, so daß dies nur dem erfahrenen Programmierer vorbehalten bleibt. Für eigene Versuche seien sie dennoch hier aufgeführt:

TE PTR STL PREPARE $0B6B
sichert die aktuelle Bildschirmzeile und blendet die Drucker-Statuszeile ein. Das aktuelle Ausgabeattribute und der aktuelle Cursor-Ticker (=Interrupt-Zähler) werden über DE zurückgegeben.

TE PTR MSG PREPARE $0B7D
bereitet, ähnlich der Funktion TE STL PREPARE (siehe oben), die Statusausgabe für den Drucker vor und ruft die Ausgabe-Routine auf.

TE PTR STL RESTORE $0BA1
schließt die Drucker-Statusausgabe ab und stellt den ursprünglichen Statuszeilen-Inhalt wieder her.

Damit zunächst genug über die Statusausgabe. Das Programm SMENU (siehe Listing) demonstriert, wie Sie Ihre neuen Betriebssystem-Kenntnisse anwenden können. Das Wechseln der Cursor- und Auswahlfelder erfolgt hierbei mit den Cursor- (Pfeil-) Tasten. Wie wäre es denn zum Beispiel mit einem Programm zur Einstellung der SIO-Parameter oder zur Änderung der Schriftarten des Druckers über die Statuszeile? Viel Spaß beim Programmieren!

Norbert Finke/rs

Hinweis:

1. Alle im Beitrag aufgeführten und verwendeten Adressen beziehen sich auf die BIOS-Version 1.4 (CP/M-Systemdatei J14GCPM3.EMS).

2. In Ermangelung offizieller Dokumentation wurden die Bezeichnungen (Namen) der beschriebenen Routinen, in Anlehnung an die ROM-Routinen der CPC-Computer, vom Autor 'erfunden'.

3. Da sämtliche hier erläuterten Routinen in der System-Speicherbank (Bank #0) liegen, muß vor dem Aufruf die Speicherbank umgeschaltet werden. Dazu läßt sich die BIOS- Funktion #30 (USERF) verwenden, die mit

```
CALL 0FC5AH
DEFW routine
```

aufgerufen werden kann. 'routine' bezeichnet dabei die Adresse der gewünschten System-Routine in der Speicherbank #0.

Alle Standard-Register werden bei der Umschaltung der Speicherbänke gesichert, so daß sie für die Übergabe von Parametern verwendet werden können.

```
;----------------------------------------------------
; SMENU.MAC - Demonstrations-Programm zur Nutzung
;    der Systemroutinen zur Ausgabe von Status-
;    meldungen auf der Statuszeile.
; Rechner:    PCW 8256/8512 (JOYCE)
; Assembler:  M80.COM (MicroSoft)
; CP/M-Vers.: CP/M 3 (PLUS) v. 1.4
; written 1990 by NoFi
;----------------------------------------------------
;EQUATES:
;
eot             equ 000ffh     ;Textende-Zeichen
bios$tbl        equ 009cah     ;Adress-Eintrag der
                               ;Statustabelle des
                               ;BIOS
bios$userf      equ 0fc5ah     ;BIOS-Einsprung
                               ;(Userf)
te_stl_prepare  equ 00b2ah     ;Einsprung System-
                               ;routine
te_stl_clear    equ 00b4dh     ;Einsprung System-
                               ;routine
bdos            equ 00005h     ;BDOS-Einsprung
```

```
strout          equ 00009h     ;BDOS-Funktion PRINT
                               ;STRING
dcio            equ 00006h     ;BDOS Direct Console
                               ;I/O
        ;
        ;zunächst die Status-Routine in die
        ;COMMON-Memory ($C000-$FFFF)
        ;übertragen
        ;
p$start:  ld    hl,(bdos+1)    ;Untere Adresse
                               ;CP/M-System
          dec   h              ;Page -1
          ld    a,h            ;prüfen, ob noch im
          cp    0c0h           ;COMMON-Memory
          jr    c,mem$err      ;- nicht genug
                               ;Speicher -
          ld    l,0            ;HL = Page-Start
          ld    (stl$rout),hl  ;für Aufruf sichern
          ld    sp,hl          ;Stack-Pointer
                               ;setzen
          ld    de,stl$out     ;Startadr. Status-
                               ;ausgabe
```

```
        ex   de,hl           ;Übertragung Vorbe-
                             ;reiten
        ld   bc,stl$end-stl$rout ;Routine/
                             ;Statusmeldung
        ldir                 ;in COMMON-Memory
                             ;übertragen
        ;
        ;dann Auswahlmaske/Cursormaske für
        ;Statusmeldung erzeugen
        ;
call$stl: ld  hl,stat$mask   ;HL auf Statusmasken
        ld   c,(hl)          ;Cursor-Maske -> C
        ld   b,4             ;Zähler für Auswahl-
                             ;maske
        xor  a               ;A = 0
gen$mask: inc hl             ;HL auf nächstes
                             ;Statusbyte
        or   (hl)            ;Status holen
        rrca                 ;und Maske erzeugen
        rrca
        djnz gen$mask        ;und weiter, bis
                             ;Maske fertig
        ld   d,a             ;Auswahl-Maske -> D
        ld   hl,(stl$rout)   ;Startadresse
                             ;Statusausgabe
        ld   l,smsg-stl$out  ;Offset zur
                             ;Statusmeldung
        ;
        ;System-Routine aufrufen (Status
        ;ausgeben)
        ;
        xor  a               ;A=0 (linke Spalte)
        ld   b,a             ;B=0 (Nr.der Status-
                             ;meldung)
        call bios$userf      ;BIOS-Funktion #30
                             ;(USERF)
stl$rout: defw $             ;Startadr. Status-
                             ;ausgabe
        ;
        ;Eingabe zur Änderung des Status bzw.
        ;Cursorfeldes von der Tastatur holen
        ;und auswerten
        ;
        ld   c,dcio          ;BDOS-Dirct Console
                             ;I/O
        ld   e,0fdh          ;Funktion = Console
                             ;input
        call bdos            ;Taste holen
        ld   hl,stat$mask    ;HL -> Cursormaske
        cp   01              ;Taste-Prüfen
        jr   z,left          ;- Pfeil links -
        cp   06
        jr   z,right         ;- Pfeil rechts -
        cp   01eh
        jr   z,down          ;- Pfeil unten -
        cp   01fh
        jr   z,up            ;- Pfeil oben -
        cp   01bh            ;EXITTaste gedrückt?
        jr   nz,call$stl     ;- kein 'EXIT' -
        ;
        ;Statusausgabe beenden, Statuszeile
        ;löschen
        ;
        call bios$userf      ;Statuszeile wieder
                             ;löschen
        defw te_stl_clear
        jp 0                 ;und Programm
                             ;beenden
        ;
        ;Fehlermeldung ausgeben, wenn Routine
        ;und Statustabelle nicht im COMMON
        ;Memory liegt
        ;
mem$err: ld  de,err$msg      ;Fehlermeldung
        ld   c,strout        ;über BDOS ausgeben
        call bdos
        jp   0               ;und Programm
                             ;beenden
        ;
        ;Cursorfeld nach links schieben
        ;
left:   ld   a,(hl)          ;Cursormaske holen
        rlca                 ;Maskenbit rotieren
        and  0f0h            ;prüfen
        jr   nz,m$save       ;-neue Maske sichern
        ld   a,010h          ;sonst neu setzen
        jr   m$save          ;und sichern
        ;
        ;Cursorfeld nach rechts schieben
        ;
right:  ld   a,(hl)          ;Cursormaske holen
        rrca                 ;Maskenbit rotieren
        and  0f0h            ;prüfen
        jr   nz,m$save       ;-neue Maske sichern
        ld   a,080h          ;sonst neu setzen
m$save: ld   (hl),a          ;Maske sichern
        jr   call$stl        ;und ausgeben
        ;
        ;Status-Auswahl -1
        ;
down:   ld   a,(hl)          ;Cursormaske holen
nxt$down: inc hl             ;und aktives Feld
                             ;suchen
        rlca
        jr   nc,nxt$down
        dec  (hl)            ;wenn gefunden,
                             ;Auswahl -1
        jp   p,call$stl      ;- o.k., Status
                             ;ausgeben -
        ld   (hl),2          ;sonst Auswahlmaske
                             ;= 2
        jr   call$stl        ;und ausgeben
        ;
        ;Status-Auswahl +1
        ;
up:     ld   a,(hl)          ;Cursormaske holen
nxt$up: inc  hl              ;und aktives Feld
                             ;suchen
        rlca
        jr   nc,nxt$up
        inc  (hl)            ;dann Auswahl +1
        ld   a,2             ;und prüfen
        cp   (hl)
        jr   nc,call$stl     ;wenn o.k., Status
                             ;ausgeben
        ld   (hl),0          ;sonst Auswahl = 0
        jr   call$stl        ;und Status ausgeben
        ;
        ;Cursor- und Auswahlmasken zur Dar-
        ;stellung und Auswahl der Status-
        ;meldungen
        ;
stat$mask: defb 080h         ;Cursorfeld-Maske
        defb 0,0,0,0         ;4 Statusbytes für
                             ;die Auswahl
                             ;(STATUS1/2/3)
        ;
        ;Fehlermeldung
        ;
err$msg: defb 'Nicht genug Speicher',10,13,'$'
        ;
        ;Diese Routine und die folgende
        ;Statustabelle müssen in die COMMON
        ;Memory verschoben werden, bevor sie
        ;aufgerufen werden können
        ;
stl$out: push hl             ;Adresse von smsg:
                             ;sichern
        ld   hl,(bios$tbl)   ;Adresse BIOS-
                             ;Statustabelle
        ex   (sp),hl         ;und gegen neue
                             ;Adresse
        ld   (bios$tbl),hl   ;austauschen
        call te_stl_prepare  ;Status-Meldung
                             ;ausgeben
        pop  hl              ;Originaladresse
                             ;wieder
        ld   (bios$tbl),hl   ;zurückspeichern
        ret                  ;und fertig
        ;
        ;Statustabelle zur Ausgabe der Status-
        ;meldungen
        ;
smsg:   ;Status-Meldung #0
        defb 081h,'-',082h,':',0f6h,'  ',0f7h,082h
        defb '1',0f5h,05,06,07,084h,082h,'2',0f5h,05
        defb 06,07,084h,082h,'3',0f5h,05,06,07,084h
        defb 082h,'4',0f5h,05,06,07,084h,088h,eot
        ;Textfragment #1
        defb 'STATUS',eot
        ;Textfragment #2
        defb 'Menue',eot
        ;Textfragment #3
        defb '/',081h,eot
        ;Textfragment #4
        defb 0f6h,020h,0eh,020h,0f7h,eot
        ;Textfragment #5
        defb 083h,'1',eot
        ;Textfragment #6
        defb 083h,'2',eot
        ;Textfragment #7
        defb 083h,'3',eot
        ;Textfragment #8
        defb '>EXIT<',eot
stl$end equ $
        end
```

Biete Software

PD-SOFTWARE -,65 bis 2,- Für IBM/ Kompat. Kat-Disk. gratis., M. Karbach, Remscheider Str. 18, 5650 Solingen 1G

CPC-MAGAZINE, 85-89, Heft DM 1,- SOFTWARE billig, 0 30/7 51 86 12

Original CPC Software ab 20 DM Jens Schneider, Tel.: 0 24 23/26 35

8000 PD & Shareware auf Lager ab DM 0,65! KATALOGDISKETTE 5.25" KOSTENLOS SONDERANGEBOT – DER PC-KAUFMANN für DM 685, EDV COMPUTER PD EXPRESS Willi Stern, Kennedystr. 38, 8039 Puchheim, Tel.: 0 89-8 00 25 45 FAX: 0 89/80 89 12 G

Biete Hardware

2 CPC464 m. GT65/CTM644/DDI-1/ Drucker/JOY/div.Prog./Liter. a. einzeln, VB 850 DM Böttcher Bernburger-Str.18, O-4020 Halle

PCW 8512 kompl. m. CPS 8256 Interface zusätzl. Scanner. Software: Locoscript 2.28, Locospell, dBase II, Prompt, Faktura, Biorhythmus, Astrologie, zahlr. Spiele und Sonderhefte. Festpreis DM 550,-. Lehmann, Tel. 0 21 04/7 11 61

CPC 464 + SP 256 + DDI-1 + DMP 2000 + 5 1/4" STARDRIVE + ENDLOSPAPIER + ca. 25 HEFTE SCHNEIDER/AMSTRAD INT. VB 550 DM TEL. 0 71 21/2 26 57 od. 2 25 92

PCW 8256 + 256KB + PCI AB '87 + SOFTWARE (TURBO + SPIELE) + BÜCHER PREIS VS, TEL. 0 23 73/45 97

CPC 6128 DPM2000 GTG5 5,25 Laufw. Joysticks ca. 30 orig. Spiele div Lit. Kompl. VB 800, Tel. 0 21 54/4 03 27 nach 18 Uhr

JOYCE PCW 8512 MIT VIEL ZUBEHÖR ZU VERKAUFEN. TEL. 0 78 05/36 55

CPC 464 grün, Vortex SP 512, 3" + 5 1/4" Floppies + dB II + WS + viele Spiele + Zubehör. VB 500,-, 0 69/67 29 14

Wegen Umstellung auf Laserdrucker 3 Stück Drucker 9512 Typenrad günstig abzugeben. Tel.: 0 57 02-99 72 (Herr Schemm)

Verkaufe CPC 6128 mit GT 65-Monitor, EPSON-Drucker RX-80F/T+ (9-Nadler), Druckerkabel und ca. 20 3"-Disketten mit reichlich Software. Unter anderem: Textverarbeitung, Universaldatei, Chemie-, Matheprog., Spiele usw. Preis: 500,- DM VHB. Tel. 04 21/58 14 90

CPC 6128 (grün) + Floppy + Printer NLQ 401, 700 DM, Tel. 0 52 51/4 95 96

Für 6128: dBase II + Buch 90 DM 5,25"-820 KB 190 DM/UWE LIESS 0 86 69/3 74 66, 13-17 Uhr

Verkaufe SCSI-Harddisk-Controller, 80-Mega-Harddisk mit Controller, installiert (Autoboot, Software), XT-Karte mit 5.25 Floppy und Modem (videotextfähig, ->Telebanking, etc.) externe 3.5 Floppy, Monitor 1084, Amiga 500. Alles zusammen oder einzeln, günstig. David Schlaepfer, Dorfstr. 80, CH-8126 Zumnikon, Tel. 01 918 10 78

3,5"-Floppy f. JOYCE, 1x84 Spur, 365 KB, m. Anschlußkabel u. Programmdiskette DM 75,-, 0 61 28/4 21 53

Suche Software

BAVARIAN/F. BOMBER: 0 68 37/79 38

Suche Joyce-Software (vor allem Jewels of Darkn./Lord o.t. Rings) T. Dahm, St. Johannerstr. 47, 6600 Saarbrücken

Suche Hardware

Suche: Graphpad III, Mouse + Programm Sonderheft 3 + 4 f. Joyce, 0 63 71/38 43

Verschiedenes

Wir räumen unser Lager für CPC-Zubehör !

- SOFTWARE (nur Disk) ab 6,- DM,
- ca. 1000 Bücher ab 4,00,
- Hardware (z.B. GT65 49,-DM),
- Zubehör und vieles mehr.
- Bestandsliste anfordern !!!

Wir liefern nur, solange der Vorrat reicht.

CLOOTS-Datentechnik
Kirchstr. 15, Tel: 0 24 51-46 60
W-5132 Übach-Palenberg G

Tausche

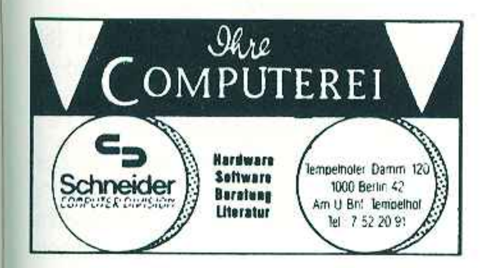
ICH SUCHE TAUSCHPARTNER FÜR CPC-DISK. LISTEN AN: MARKUS MACHEREY, RINNEBACHSTR. 24, 5166 KREUZAU-4, TEL. 0 24 22/87 37

Club

• CPC POWER International 8/91 • Die Franzine für den CPC/Plus!! Jetzt wieder da! 5,- DM bar/Scheck. Sofort wieder bestellen CPC POWER, PF 501132, 5 KÖLN 50, 30 DIN A4-Seiten stark, anwenderorientiert!

Händlerverzeichnis

Berlin



Einträge möglich mindesten 6 x innerhalb eines Insertionsjahres.

Eintragungen im Händlerverzeichnis, nach Städten geordnet, kosten je mm Höhe 6,– DM bei einer Spaltenbreite von 58 mm.

Löhne/Ostwestfalen



Computer- & Softwarezentrum für Norddeutschland
AMSTRAD-, SCHNEIDER- & VORTEX Regionalhändler & **SERVICE-CENTRALE.** Sämtl. Computer, Drucker, Peripherie & Zubehör v. A–Z, EDV Papier etc. • Discs
Fritz OBERMEIER COMPUTER • TELEFAX • BTX • HIFI • VIDEO • TV • – NEC • EPSON • TANDON • BROTHER • SEIKO • OKI • STAR • LOCO • etc.
am Bahnhof • Bünder Straße 20 • 4972 LÖHNE 1 • Tel. 0 57 32 61 26/32 46

Nähere Informationen:
DMV-Verlag
Sylvia Stephani
Telefon (0 56 51) 8 09-380



Ein Hauch von Lexus

Lexikon die Dritte

Das Jetsam-Programm "Mini-Lexikon" hat es verdient, ein weiteres Mal in verjüngter und verbesserter Form an die Öffentlickeit zu treten. Es möchte sich denjenigen vorstellen, die nicht auf Qualität, Komfort und Übersichtlichkeit verzichten wollen und dafür gerne bereit sind, etwas mehr Speicherplatz zu opfern. Aber auch das Speicherplatzproblem ist eigentlich keines mehr – dank der Flexibilität des Programms.

Mit dem "Lexikon" können Sie Informationen zu den unterschiedlichsten Themen (zum Beispiel: Rezepte, Rockmusik etcetera) archivieren, wie Sie es von Ihrem Lexikon her kennen. Sie geben das Stichwort und den dazugehörigen Erläuterungstext ein. Alles weitere entnehmen Sie bitte den folgenden Erklärungen.

Das Programm ist in vier Menüs gegliedert:

1. das Hauptmenü mit dem Optionenmenü
2. das Printermenü
3. das Filesmenü
4. das Diskettenmenü

Da alle Menüs über einen eigenen Hilfstext verfügen, dürfte auch Ihren Kindern der Umgang mit dem Programm nicht schwerfallen; deshalb sei dazu nur das Nötigste erklärt:

Das Hauptmenü und das Optionenmenü

<E>ingabe: Hier können Sie Ihre Datensätze eingeben. Jede Zeile muß mit <RETURN>, der gesamte Satz mit Gitter(#) beendet werden. Es empfiehlt sich, wie von Textprogrammen gewohnt, ab und zu zwischenzuspeichern. Sie können durch erneutes Anwählen unter dem gleichen Stichwort mit der Eingabe fortfahren. Der Datensatz wird dadurch nicht überschrieben.

<S>uchen: Nach Eingabe des Stichwortes wird der betreffende Datensatz auf dem Monitor angezeigt.

<K>orrektur: Nach Eingabe eines Stichwortes kann der Datensatz Zeile für Zeile durch Neueingabe korrigiert werden. Sind alle betreffenden Zeilen berichtigt, kann die Bearbeitung beendet und der geänderte Satz überprüft werden.

<L>öschen: Nach Eingabe des Stichwortes und Bestätigung der Sicherheitsabfrage wird der betreffende Datensatz gelöscht.

<P>rinter, <F>iles, <D>iskette und <O>ptionen führen Sie jeweils in ein neues Menü.

E<x>it: Dieser Punkt steht Ihnen in allen Menüs zur Verfügung. Das Programm wird damit beendet und kehrt zu Basic zurück. Zuvor wird jedoch die geöffnete Datei geschlossen.

Das Optionenmenü enthält folgende Punkte:

<I>ndex: Der Index ist ebenfalls bei den Funktionen <S>uchen, <K>orrektur und <L>öschen abrufbar. Alle Stichworte werden in alphabetischer Reihenfolge auf dem Monitor angezeigt.

<L>exikon: Alle Datensätze werden in alphabetischer Reihenfolge auf dem Monitor angezeigt. Die Ausgabe kann jederzeit beendet werden.

<C>lear: Auch dieser Punkt steht Ihnen in allen Menüs zur Verfügung. Das Programm kehrt in das Ausgangsmenü (in der Regel das Hauptmenü) zurück.

Das Printermenü

Es ist das einzige Menü, bei dem Sie das aktuelle Laufwerk und die aktuelle Datei nicht mehr im Blickfeld haben. Statt dessen können Sie wählen, ob der Ausdruck als <E>ntwurf, <K>orrespondenz oder in <M>icroschrift erfolgen soll.

<S>uchen: Nach der Stichworteingabe wird der betreffende Datensatz ausgedruckt.

<I>ndex: Das Inhaltsverzeichnis wird alphabetisch ausgedruckt.

<L>exikon: Alle Datensätze werden in alphabetischer Reihenfolge ausgedruckt. Der Vorgang kann jederzeit gestoppt werden.

<W>ählen: Der Datensatz wird Ihnen nach der Stichworteingabe auf dem Monitor angezeigt. Erst dann bestimmen Sie, ob er ausgedruckt wird.

<Z>eigen: Hiermit können Sie das gesamte Lexikon "durchblättern", wenn Sie zum Beispiel Datensätze zu einem bestimmten Thema ausdrucken lassen wollen.

Das Filesmenü

Möchten Sie wissen, welche Dateien das Programm zuladen kann? Kein Problem! Beim Anwählen des Filesmenüs zeigt es Ihnen selbsttätig die Lexikon-

Dateien des mit der Variablen "lw$" bestimmten Laufwerks an.
Dateien <A><B><M>: Hiermit können Sie auch die anderen Laufwerke einsehen. Das Standardlaufwerk bleibt dabei bestehen.
<L>öschen: Nach Bestätigung der Sicherheitsabfrage wird die aktuelle Datei gelöscht.
<U>mbenennen: Nach Eingabe des neuen Namens wird die aktuelle Datei mit dem neu einzugebenden Namen versehen.
<W>echseln: Sie können die Datei wechseln oder eine neue einrichten.

Das Diskettenmenü

Beim Anwählen des Menüs zeigt Ihnen das Programm selbsttätig alle gespeicherten Programme und Dateien auf Ihrer Diskette an.
Directory <A><B><M>: Durch Druck der entsprechenden Taste können Sie auch die anderen Laufwerke einsehen.
<D>iskette: Haben Sie eine neue Diskette in das angemeldete Laufwerk eingelegt, so können Sie diese hiermit anmelden.
<L>aufwerk: Hier wechseln Sie das Laufwerk. Alle Dateioperationen erfolgen nun auf dem neu angemeldeten Laufwerk.
<C>lear: Die Clear-Funktion ist im Diskettenmenü gesperrt, wenn Sie zuvor im Filesmenü eine Datei gelöscht haben.
Bevor Sie das Listing eingeben, sollten Sie etwas über die Besonderheiten des Programms erfahren, die Ihnen die Arbeit angenehmer machen sollen.

Zur Eingabe

1. Beim Start des Programms wird mit dem Befehl "OPTION FILES lw$" das Laufwerk A zum Standardlaufwerk und damit zum Speichern Ihrer Daten bestimmt. Möchten Sie Ihre Dateien auf Laufwerk B speichern, ändern Sie bitte in Zeile 5560 die Variable "lw$" in: lw$="B".
2. Im Gegensatz zu den Vorgängern dieses Programms ist die Stichworteingabe auf 15 Zeichen (statt 20) begrenzt. Wollen Sie Daten aus den vorigen Programmen verarbeiten, müssen Sie in den folgenden Zeilen jeweils 15 durch 20 ersetzen: 560, 590, 600, 1490, 1540, 1550 u. 4160.
Zudem müssen folgende Zeilen geändert werden:
1740 PRINT;w$;" ";CHR$(149);
1760 ....MOD 50 in MOD 40 ändern
2360 LPRINT;w$;" * ";
2380/2400 ....MOD 5 in MOD 4 ändern
Bitte vergessen Sie nicht, Ihre alten Dateien in die Dateitypen .LEX und .IND umzubenennen!
3. Da Jetsam-Dateien stets geschlossen werden müssen, ist dieses Programm zweifach gesichert. Mit dem Befehl "OPTION RUN" ist die Stoptaste wirkungslos; ein "ON ERROR GOTO" in Zeile 90 sorgt dafür, daß Ihre Dateien, sollte wider Erwarten ein Fehler auftreten, stets geschlossen werden. Das hat den Nachteil, daß Sie von Basic keine Fehlermeldungen erhalten, wenn Ihr abgetipptes Listing nicht korrekt ist. Nehmen Sie deshalb diesen Befehl heraus, bis Ihr Programm einwandfrei läuft. Denken Sie im Falle eines Falles aber bitte daran, Ihre Datei mit "CLOSE #1" zu schließen, bevor Sie einen Fehler editieren.
Bleibt noch zu sagen, daß der Platz hier leider nicht ausreicht, um noch die Fehlerbehandlungen und einige Besonderheiten des Programms zu beschreiben. Da es jedoch sehr übersichtlich ist, finden Sie alles weitere sicher selbst heraus.
Die Startdiskette ist sehr einfach erstellt. Sie benötigen Ihre Systemdatei J??GCPM3.EMS, SUBMIT.COM, BASIC.COM, LEXIKON.BAS und für die Arbeit mit Endlospapier die Datei PAPER.COM. Mit RPED.BAS erstellen Sie dann folgende PROFILE.SUB:

```
paper 12
basic lexikon
```

Wenn Sie alles auf Ihre Startdiskette kopiert haben, kann es bereits losgehen. Viel Vergnügen!!!

(Wolfgang Horstmann/rs)



```
LISTING >LEXIKON <, REMARK = >'<.

< 1> 10 ' *******************
< 2> 20 ' *.. LEXIKON.BAS.. *
< 3> 30 ' *.... 06/1991.... *
< 4> 40 ' * by W. Horstmann *
< 5> 50 ' * & CPC Internat. *
< 6> 60 ' *******************
< 7> 70 '
<33> 80 GOSUB 5490 'Initialisierung und Titel
<36> 90 OPTION RUN
<51> 100 ON ERROR GOTO 5200
<20> 110 '
< 5> 120 PRINT FNloc$(26,0);line$
<27> 130 PRINT;TAB(16);invon$;".... Bitte die Date
     ndiskette in Laufwerk ";lw$;" einlegen!.... "
     ;invoff$;line$
<13> 140 taste$=INPUT$(1)
<71> 150 RESET
<55> 160 OPTION FILES lw$
<32> 170 '
<96> 180 GOSUB 3450 'Dateien anzeigen
<79> 190 GOSUB 3970 'Datei wählen und öffnen
< 9> 200 GOTO.. 410 'Hauptmenue
<21> 210 '
<36> 220 '----- UP Auf Antwort warten -----
<25> 230 '
```

```
<26> 240 antwort$=INKEY$:IF antwort$="" THEN GOTO
     240
<11> 250 antwort$=UPPER$(antwort$):antwort=INSTR(m
     atch$,antwort$)
<57> 260 IF antwort=0 THEN GOTO 240
<82> 270 RETURN
<35> 280 '
<28> 290 '----- UP Auf Taste warten -----
<20> 300 '
< 5> 310 PRINT FNloc$(26,0);line$
<22> 320 PRINT FNinfo$("Weiter mit beliebiger Tast
     e!")
<13> 330 taste$=INPUT$(1)
<77> 340 RETURN
<30> 350 '
<32> 360 '
<34> 370 ' *******************
<56> 380 '..... Hauptmenue....
<38> 390 ' *******************
<21> 400 '
<42> 410 PRINT FNloesch$
<92> 420 PRINT FNloc$(2,0);SPC(15);"Laufwerk: ";lw
     $;SPC(30);"Datei: "+datfile$
< 1> 430 PRINT FNloc$(4,0)+invon$+" <E>ingabe <S>u
     chen <K>orrektur <L>oeschen <P>rinter <F>iles
      <D>iskette <O>ptionen E<x>it "+invoff$
<55> 440 match$="ESKLPFDXO":GOSUB 240
```

```
<75> 450 ON antwort GOTO 550,840,1020,1280,2090,31
     90,4240,5230
<33> 460 '
<59> 470 '----- Optionen-Menue -----
<37> 480 '
<35> 490 PRINT FNloc$(4,0)+invon$+"..............
     ......... <I>ndex <L>exikon <H>elp E<x>it <C>
     lear........................ "+invoff$
<37> 500 match$="ILHXC":GOSUB 240
<44> 510 ON antwort GOTO 1620,1860,4780,5230,430
<26> 520 '
<28> 530 ' ***** EINGABE *****
<30> 540 '
<95> 550 PRINT FNloc$(4,1)" Eingabe "
<82> 560 PRINT FNloc$(7,0)"Stichwort: ";STRING$(15
     ,46);SPC(20)
<27> 570 PRINT FNloc$(7,10);con$;:INPUT" ",xy$:PRI
     NT coff$
<16> 580 IF xy$="" THEN PRINT FNloesch$:GOTO 430 '
     Menue
< 3> 590 IF LEN (xy$)>15 THEN PRINT bell$:GOTO 550
<80> 600 xy$=UPPER$(xy$):xy$=xy$+SPACE$(15-LEN(xy$
     ))
<25> 610 e=SEEKKEY(#1,0,0,xy$)
<55> 620 PRINT FNloc$(6,0);xy$:PRINT line$
<21> 630 zeile=7
<81> 640 PRINT FNloc$(zeile,0);STRING$(90,46)
<78> 650 PRINT FNloc$(zeile,0);con$:LINE INPUT xs$
<45> 660 IF xs$="#" THEN GOTO 720
<13> 670 LSET w$=xy$:LSET s$=xs$
<75> 680 e=ADDREC(#1,2,0,w$)
<79> 690 PUT #1:zeile=zeile+1
< 1> 700 IF zeile=26 THEN GOTO 770 ELSE GOTO 640
<26> 710 '
<10> 720 erg=CONSOLIDATE(#1)
<25> 730 PRINT cup$;cup$;line$
<44> 740 PRINT coff$:GOSUB 310 'Taste
<26> 750 PRINT FNloesch$:GOTO 430 'Menue
<36> 760 '
<19> 770 PRINT FNloc$(26,0);coff$;line$
< 6> 780 PRINT FNinfo$("Bitte zur weiteren Eingabe
      <RETURN> druecken!")
<29> 790 taste$=INPUT$(1)
<31> 800 IF taste$=CHR$(13) THEN PRINT FNloc$(8,0)
     ;loesch$:GOTO 630 ELSE GOTO 790
<27> 810 '
<29> 820 ' ***** SUCHEN *****
<31> 830 '
<49> 840 PRINT FNloc$(4,11)" Suchen "
<86> 850 GOSUB 1420 'Indexabfrage, Stichworteingab
     e
<65> 860 PRINT FNloc$(6,0);xy$:PRINT line$
<75> 870 PRINT FNloc$(8,0);loesch$;cup$
<45> 880 WHILE e=0
<58> 890.. GET #1
<32> 900.. PRINT s$
<20> 910.. suchen=suchen+1
<23> 920.. e=SEEKNEXT(#1,0)
< 0> 930.. IF 0=(suchen MOD 19) THEN GOSUB 310:GOT
     O 870 'Taste
<95> 940 WEND
<72> 950 PRINT line$:suchen=0
<63> 960 IF druck=1 THEN druck=0:RETURN
<50> 970 IF lexikon=1 THEN lexikon=0:RETURN
<39> 980 GOSUB 310:PRINT FNloesch$:GOTO 430 'Taste
     - Menue
<44> 990 '
<83> 1000 ' ***** KORREKTUR ******
<86> 1010 '
<51> 1020 PRINT FNloc$(4,20)" Korrektur "
< 2> 1030 GOSUB 1420 'Indexabfrage, Stichworteinga
     be
<76> 1040 PRINT FNloesch$:PRINT "Bearbeitet wird:
     ";xy$
<36> 1050 WHILE e=0
<14> 1060.. GET #1
<67> 1070.. PRINT FNloc$(9,0);loesch$:PRINT s$
<60> 1080.. PRINT FNloc$(15,0)"Diesen Satz neu ein
     geben? <J/N>";SPC(10)
<50> 1090.. match$="JN":GOSUB 240:PRINT cup$;loesc
     h$
<19> 1100.. IF antwort=2 THEN GOTO 1180
<33> 1110.. PRINT FNloc$(11,0);STRING$(90,46)
<31> 1120.. PRINT FNloc$(11,0);con$:LINE INPUT xs$
<67> 1130.. IF xs$="" THEN PRINT coff$:GOTO 1150
<28> 1140.. LSET s$=xs$:PUT #1:PRINT coff$
<94> 1150.. PRINT FNloc$(15,0)"Weiter mit <SPACE>.
     *. Beenden mit <CAN>"
<49> 1160.. taste$=INPUT$(1):PRINT cup$;loesch$
<63> 1170.. IF taste$=CHR$(8) THEN GOTO 1200
< 1> 1180.. e=SEEKNEXT(#1,0)
<40> 1190 WEND
< 9> 1200 erg=CONSOLIDATE(#1)
<96> 1210 PRINT FNloc$(15,0)"Soll der Datensatz ue
     berprueft werden? <J/N>";SPC(10)
<95> 1220 match$="JN":GOSUB 240:PRINT FNloesch$
<93> 1230 IF antwort=2 THEN GOTO 430 'Menue
<28> 1240 e=SEEKKEY(#1,0,0,xy$):GOTO 860 'Datensat
     z anzeigen
< 3> 1250 '
< 6> 1260 ' ***** LOESCHEN ******
< 9> 1270 '
<72> 1280 PRINT FNloc$(4,32)" Loeschen "
<24> 1290 GOSUB 1420 'Indexabfrage, Stichworteinga
     be
<26> 1300 PRINT"Soll der Datensatz '";xy$;"' geloe
     scht werden? <J/N>"
<94> 1310 match$="JN":GOSUB 240:PRINT cup$;loesch$
<60> 1320 IF antwort=2 THEN GOSUB 1490:GOTO 1300 '
     Neue Eingabe
< 3> 1330 WHILE e=0 OR e=101
<38> 1340.. erg=DELKEY(#1,0)
<65> 1350.. e=SEEKKEY(#1,0,0,xy$)
<35> 1360 WEND
<32> 1370 erg=CONSOLIDATE(#1)
<72> 1380 PRINT FNloesch$:GOTO 430 'Menue
<17> 1390 '
< 9> 1400 '----- UP Indexabfrage; Stichwort eingeb
     en, pruefen, bearbeiten -----
<94> 1410 '
<25> 1420 e=SEEKRANK(#1,0,0)
<63> 1430 IF e<>0 THEN GOSUB 5340:GOTO 420 'Fehler
     meldung - Hauptmenue
<61> 1440 PRINT FNloc$(7,0)"Index anzeigen? <J/N>"
< 9> 1450 match$="JN":GOSUB 240:PRINT FNloesch$
< 8> 1460 ON antwort GOSUB 1700 'Index anzeigen
<13> 1470 '
<71> 1480 IF index=0 THEN z=7 ELSE index=0:z=22
< 0> 1490 PRINT FNloc$(z,0)"Stichwort eingeben: "+
     STRING$(15,46);SPC(20)
<64> 1500 PRINT FNloc$(z,19);con$;:INPUT" ",xy$:PR
     INT coff$
<24> 1510 IF xy$<>"" THEN GOTO 1540 ELSE PRINT FNl
     oesch$
<47> 1520 IF druck=0 THEN GOTO 430 'Menue
<40> 1530 druck=0:GOTO 2100 'Printermenue
<96> 1540 IF LEN(xy$)>15 THEN PRINT bell$:GOTO 149
     0
<11> 1550 xy$=UPPER$(xy$):xy$=xy$+SPACE$(15-LEN(xy
     $))
<47> 1560 e=SEEKKEY(#1,0,0,xy$)
<91> 1570 IF e<>0 THEN GOSUB 5390:GOTO 1490 'Fehle
     rmeldung
<13> 1580 RETURN
<21> 1590 '
<95> 1600 ' ***** INDEX *****
<98> 1610 '
<12> 1620 PRINT FNloc$(4,25)" Index "
< 5> 1630 GOSUB 1680:index=0 'Index anzeigen
<80> 1640 GOSUB 310:PRINT FNloesch$:GOTO 490 'Tast
     e - Optionen-Menue
<11> 1650 '
<42> 1660 '----- UP Index anzeigen -----
<17> 1670 '
<47> 1680 e=SEEKRANK(#1,0,0)
<22> 1690 IF e<>0 THEN GOSUB 5340:GOTO 430 'Fehler
     meldung - Hauptmenue
<13> 1700 PRINT FNloc$(6,0);line$
<11> 1710 PRINT FNinfo$("* * * * *. I N D E X. * *
      * * *")
< 8> 1720 WHILE e=0 OR e=101
<92> 1730.. index=index+1:GET #1
<75> 1740.. PRINT "* ";w$;" ";
< 8> 1750.. e=SEEKSET(#1,0)
<58> 1760.. IF 0=(index MOD 50) THEN GOTO 1810
<46> 1770 WEND
<51> 1780 PRINT FNloc$(19,0);line$
<20> 1790 RETURN
< 0> 1800 '
<10> 1810 PRINT line$:GOSUB 310 'Taste
<53> 1820 PRINT FNloc$(10,0);loesch$:GOTO 1700
< 9> 1830 '
<12> 1840 ' ***** LEXIKON *****
<15> 1850 '
<91> 1860 PRINT FNloc$(4,33)" Lexikon "
<48> 1870 e=SEEKRANK(#1,0,0)
<23> 1880 IF e<>0 THEN GOSUB 5340:GOTO 430 'Fehler
     meldung - Hauptmenue
<31> 1890 WHILE e=0 OR e=101
<14> 1900.. GET #1
```

```
<47> 1910.. PRINT FNloc$(6,0);w$:PRINT line$
<87> 1920.. e=SEEKKEY(#1,0,0,w$)
<31> 1930.. lexikon=1:GOSUB 870 'Datensätze anzeig
     en
<66> 1940.. PRINT FNloc$(26,0);line$
<50> 1950.. PRINT FNinfo$("Weiter mit <SPACE>. *.
     Beenden mit <CAN>")
< 4> 1960.. taste$=INPUT$(1)
<18> 1970.. IF taste$=CHR$(8) THEN GOTO 2020
<53> 1980 WEND
<17> 1990 PRINT FNloc$(26,0);line$
<36> 2000 PRINT FNinfo$("* * * * *. ENDE DES LEXIK
     ONS * * * * *")
<55> 2010 taste$=INPUT$(1)
<77> 2020 PRINT FNloesch$:GOTO 490 'Optionen-Menue
<93> 2030 '
<96> 2040 '
< 0> 2050 ' *******************
<50> 2060 '.... Printermenue
< 6> 2070 ' *******************
< 9> 2080 '
<32> 2090 PRINT FNloc$(2,0);SPC(15);"<E>ntwurf "+i
     nvon$+nor$+invoff$+".... <K>orrespondenz "+in
     von$+kor$+invoff$+".... <M>icroschrift "+invo
     n$+mic$+invoff$
<95> 2100 PRINT FNloc$(4,0)+invon$+"........... <S
     >uchen <I>ndex <L>exikon <W>ählen <Z>eigen <H
     >elp E<x>it <C>lear.......... "+invoff$
<63> 2110 match$="SILWZEKMHXC":GOSUB 240
<44> 2120 ON antwort GOTO 2160,2300,2460,2670,2760
     ,3010,3060,3100,4790,5230,420
<95> 2130 '
<98> 2140 ' ***** SUCHEN *****
< 2> 2150 '
<75> 2160 PRINT FNloc$(4,12)" Suchen "
<64> 2170 druck=1:GOSUB 1420 'Indexabfrage, Stichw
     orteingabe
<29> 2180 LPRINT strich$:LPRINT dop$;xy$:LPRINT st
     rich$
<51> 2190 WHILE e=0
< 1> 2200.. GET #1
<37> 2210.. LPRINT s$
<85> 2220.. e=SEEKNEXT(#1,0)
<25> 2230 WEND
<94> 2240 LPRINT strich$:LPRINT
<35> 2250 IF ausdruck=1 THEN ausdruck=0:RETURN
<91> 2260 PRINT FNloesch$:GOTO 2100 'Menue
<10> 2270 '
<13> 2280 ' ***** INDEX *****
<16> 2290 '
<12> 2300 PRINT FNloc$(4,21)" Index "
<21> 2310 e=SEEKRANK(#1,0,0)
<59> 2320 IF e<>0 THEN GOSUB 5340:GOTO 420 'Fehler
     meldung - Hauptmenue
<32> 2330 LPRINT strich$:LPRINT dop$"..... * INDEX
     DER DATEI: "datfile$" *..... ":LPRINT strich
     $
< 7> 2340 WHILE e=0 OR e=101
<43> 2350.. worte=worte+1:GET #1
<44> 2360.. LPRINT"* ";w$;" ";
< 7> 2370.. e=SEEKSET(#1,0)
< 8> 2380.. IF 0=(worte MOD 5) THEN LPRINT
<45> 2390 WEND
< 6> 2400 IF 0=(worte MOD 5) THEN LPRINT strich$:L
     PRINT:LPRINT:GOTO 2420
<84> 2410 LPRINT:LPRINT strich$:LPRINT:LPRINT
<70> 2420 worte=0:GOTO 2100 'Menue
< 2> 2430 '
< 5> 2440 ' ***** LEXIKON *****
< 8> 2450 '
<96> 2460 PRINT FNloc$(4,29)" Lexikon "
<41> 2470 e=SEEKRANK(#1,0,0)
<79> 2480 IF e<>0 THEN GOSUB 5340:GOTO 420 'Fehler
     meldung - Hauptmenue
< 8> 2490 PRINT FNloc$(26,0);line$
<50> 2500 PRINT FNinfo$("Beenden des Ausdrucks mit
     <CAN>")
<69> 2510 LPRINT strich$:LPRINT dop$".... * LEXIKO
     N DER DATEI: "datfile$" *.... "
<94> 2520 LPRINT strich$:LPRINT
< 8> 2530 WHILE e=0 OR e=101
<19> 2540.. GET #1
<50> 2550.. LPRINT strich$:LPRINT dop$;w$:LPRINT s
     trich$
<92> 2560.. e=SEEKKEY(#1,0,0,w$)
<21> 2570.. ausdruck=1:GOSUB 2190 'Datensätze druc
     ken
<32> 2580.. taste$=INKEY$
<79> 2590.. IF taste$=CHR$(8) THEN GOTO 2630
<24> 2600 WEND
<81> 2610 LPRINT strich$
<95> 2620 LPRINT dop$".... * E N D E. D E S. L E X
     I K O N S *.... ";strich$
<90> 2630 PRINT FNloesch$:GOTO 2100 'Menue
< 9> 2640 '
<12> 2650 ' ***** WAEHLEN *****
<15> 2660 '
<14> 2670 PRINT FNloc$(4,39)" Wählen "
<84> 2680 druck=1:GOSUB 850 'Index, Stichwort, Mon
     itoranzeige
<12> 2690 PRINT FNloc$(26,0);line$
<72> 2700 PRINT FNinfo$("Datensatz ausdrucken? <J/
     N>")
<42> 2710 match$="JN":GOSUB 240
<79> 2720 IF antwort=2 THEN PRINT FNloesch$:GOTO 2
     100 'Menue
<43> 2730 e=SEEKKEY(#1,0,0,xy$)
<45> 2740 GOTO 2180 'Ausdruck
<14> 2750 '
<17> 2760 ' ***** ZEIGEN *****
<20> 2770 '
<78> 2780 PRINT FNloc$(4,48)" Zeigen "
<53> 2790 e=SEEKRANK(#1,0,0)
<63> 2800 IF e<>0 THEN GOSUB 5340:GOTO 420 'Fehler
     meldung - Hauptmenue
< 8> 2810 WHILE e=0 OR e=101
<19> 2820.. GET #1
<52> 2830.. PRINT FNloc$(6,0);w$:PRINT line$
<92> 2840.. e=SEEKKEY(#1,0,0,w$)
<66> 2850.. druck=1:GOSUB 870 'Datensatz anzeigen
<71> 2860.. PRINT FNloc$(26,0);line$
<46> 2870.. PRINT FNinfo$("Weiter mit <SPACE> * Au
     sdrucken mit <COPY> * Beenden mit <CAN>")
< 9> 2880.. taste$=INPUT$(1)
<51> 2890.. IF taste$=CHR$(8) THEN GOTO 2970
<62> 2900.. IF taste$=CHR$(23) THEN e=SEEKKEY(#1,0
     ,0,w$) ELSE GOTO 2930
<46> 2910.. LPRINT strich$:LPRINT dop$;w$:LPRINT s
     trich$
<16> 2920.. ausdruck=1:GOSUB 2190 'Datensatz ausdr
     ucken
<39> 2930 WEND
< 3> 2940 PRINT FNloc$(26,0);line$
<95> 2950 PRINT FNinfo$("* * * * *. ENDE DES LEXIK
     ONS. * * * * *")
<88> 2960 taste$=INPUT$(1)
<37> 2970 PRINT FNloesch$:GOTO 2100 'Menue
<27> 2980 '
<30> 2990 ' ***** DRUCKSCHRIFTEN *****
<85> 3000 '
<42> 3010 IF nor$="ein" THEN 2110 'Menue
<11> 3020 IF kor$="ein" THEN kor$="aus":LPRINT kor
     off$
<27> 3030 IF mic$="ein" THEN mic$="aus":LPRINT mic
     off$
<31> 3040 nor$="ein":GOTO 2090 'Menue
< 1> 3050 '
<59> 3060 nor$="aus":IF kor$="ein" THEN 2110 'Menu
     e
<39> 3070 IF mic$="ein" THEN mic$="aus":LPRINT mic
     off$
<48> 3080 kor$="ein":LPRINT koron$:GOTO 2090 'Menu
     e
<13> 3090 '
<23> 3100 nor$="aus":IF mic$="ein" THEN 2110 'Menu
     e
<10> 3110 IF kor$="ein" THEN kor$="aus":LPRINT kor
     off$
<83> 3120 mic$="ein":LPRINT micon$:GOTO 2090 'Menu
     e
<96> 3130 '
< 0> 3140 '
< 3> 3150 ' *******************
< 1> 3160 '..... Filesmenue
< 9> 3170 ' *******************
<12> 3180 '
<72> 3190 PRINT FNloc$(2,0);SPC(15);"Laufwerk: ";l
     w$;SPC(30);"Datei: "+datfile$
<70> 3200 PRINT FNloc$(4,0)+invon$+"...... Dateien
     <A><B><M> * <L>oeschen <U>mbenennen <W>echse
     ln <H>elp E<x>it <C>lear...... "+invoff$
< 9> 3210 GOSUB 3450 'Dateien anzeigen
<54> 3220 match$="ABMLUWHXC":GOSUB 240
<91> 3230 ON antwort GOTO 3280,3290,3300,3540,3680
     ,3910,4800,5230
<27> 3240 PRINT FNloesch$:GOTO 430
< 5> 3250 '
< 8> 3260 ' ***** DATEIEN A/B/M ******
<11> 3270 '
<80> 3280 lwx$="A":GOTO 3310
```

```
<95> 3290 lwx$="B":GOTO 3310
<22> 3300 lwx$="M"
<15> 3310 IF lwx$=lw$ THEN PRINT bell$:GOTO 3220 '
     Menue
<90> 3320 CLOSE #1:OPTION FILES lwx$ 'Laufwerk wec
     hseln
<69> 3330 IF lwx$="M" THEN GOTO 3370
<91> 3340 PRINT FNloc$(26,0);line$
<46> 3350 PRINT FNinfo$("Bitte pruefen, ob eine Di
     skette eingelegt ist!")
<77> 3360 taste$=INPUT$(1)
<93> 3370 GOSUB 3450:GOSUB 310 'Dateien anzeigen -
      Taste
<25> 3380 PRINT FNloesch$:PRINT FNloc$(26,0);line$
<75> 3390 PRINT FNinfo$("B I T T E.. W A R T E N !
     ")
<75> 3400 OPTION FILES lw$:lwx$=lw$ 'Laufwerk zuru
     eck
<53> 3410 GOSUB 4140:GOTO 3190 'Datei oeffnen - Me
     nue
< 0> 3420 '
< 1> 3430 '----- UP Dateien anzeigen -----
< 6> 3440 '
<52> 3450 PRINT FNloesch$
<79> 3460 PRINT line$:PRINT "Dateien auf Laufwerk:
      ";lwx$:PRINT line$
<73> 3470 IF FIND$("*.LEX")="" THEN PRINT "Es sind
      noch keine Dateien eingerichtet!":GOTO 3490
<70> 3480 DIR *.LEX
<29> 3490 PRINT FNloc$(13,0);line$
<90> 3500 RETURN
<98> 3510 '
< 2> 3520 ' ***** LOESCHEN *****
< 5> 3530 '
<58> 3540 PRINT FNloc$(4,27)" Loeschen "
<53> 3550 PRINT FNloc$(16,0)"Soll die Datei "+datf
     ile$+" gelöscht werden? <J/N>"
<54> 3560 match$="JN":GOSUB 240
< 8> 3570 IF antwort=2 THEN PRINT FNloc$(15,0);loe
     sch$:GOTO 3200 'Menue
<86> 3580 CLOSE #1:KILL datfile$:KILL indfile$:spe
     rre=1
<65> 3590 PRINT FNloc$(2,0)+lwtext$
<14> 3600 GOSUB 3450 'Dateien anzeigen
<13> 3610 PRINT FNloc$(16,0)"Soll die Diskette ode
     r das Laufwerk gewechselt werden? <J/N>"
<44> 3620 match$="JN":GOSUB 240
< 3> 3630 ON antwort GOTO 4250 'Diskettenmenue
<98> 3640 GOSUB 3970:PRINT FNloc$(15,0);loesch$:GO
     TO 3190 'Datei laden - Menue
<13> 3650 '
<16> 3660 ' ***** UMBENENNEN *****
<19> 3670 '
<24> 3680 PRINT FNloc$(4,38)" Umbenennen "
<54> 3690 PRINT FNloc$(16,0)"ALTER NAME: ";file$
<64> 3700 PRINT FNloc$(17,0)"NEUER NAME: ";STRING$
     (8,46);SPC(20)
< 6> 3710 PRINT FNloc$(17,11);con$;:INPUT " ",name
     $:PRINT coff$
<97> 3720 IF name$="" THEN PRINT FNloc$(15,0);loes
     ch$:GOTO 3200 'Menue
<51> 3730 name$=UPPER$(name$)
<57> 3740 IF RIGHT$(name$,4)=".LEX" THEN PRINT bel
     l$:GOTO 3700
< 5> 3750 IF LEN(name$)>8 THEN PRINT bell$:GOTO 37
     00
<14> 3760 datname$=name$+SPACE$(8-LEN(name$))+".LE
     X":indname$=name$+".IND"
<23> 3770 IF FIND$(datname$)<>"" THEN GOTO 3850
<60> 3780 PRINT:PRINT"Soll die Datei '"+datfile$+"
     ' in '"+datname$+"' umbenannt werden? <J/N>"
<67> 3790 match$="JN":GOSUB 240
<77> 3800 IF antwort=2 THEN PRINT FNloc$(18,0);loe
     sch$:GOTO 3700
<98> 3810 CLOSE #1:NAME datfile$ AS datname$:NAME
     indfile$ AS indname$
<51> 3820 file$=name$:datfile$=datname$:indfile$=i
     ndname$
<12> 3830 GOSUB 4140:PRINT FNloc$(15,0);loesch$:GO
     TO 3190 'Datei oeffnen - Menue
<14> 3840 '
< 5> 3850 PRINT FNloc$(26,0);line$
<87> 3860 PRINT FNinfo$("Unter dem eingegebenen Na
     men existiert bereits eine Datei!")
< 0> 3870 taste$=INPUT$(1):PRINT FNloc$(26,0);loes
     ch$:GOTO 3700
<26> 3880 '
<29> 3890 ' ***** WECHSELN *****
< 4> 3900 '
<42> 3910 PRINT FNloc$(4,51)" Wechseln "
<38> 3920 CLOSE #1:GOSUB 3970 'Datei laden
<92> 3930 PRINT FNloc$(15,0);loesch$:GOTO 3190 'Me
     nue
<16> 3940 '
<17> 3950 '----- UP Datei laden bzw einrichten und
      oeffnen -----
<22> 3960 '
<73> 3970 sperre=0
<76> 3980 PRINT FNloc$(15,0);line$
<48> 3990 PRINT FNloc$(16,0)"Bitte den Namen der g
     ewuenschten oder der neuen Datei eingeben. We
     nn die Bearbeitung hier"
<60> 4000 PRINT FNloc$(18,0)"abgebrochen werden so
     ll, einfach <RETURN> druecken! ";STRING$(8,46
     );SPC(20)+line$
<21> 4010 PRINT FNloc$(18,51);con$;:INPUT" ",file$
     :PRINT coff$
<95> 4020 IF file$="" THEN GOTO 5230 ELSE file$=UP
     PER$(file$) 'Programmende
<96> 4030 IF RIGHT$(file$,4)=".LEX" THEN PRINT bel
     l$:GOTO 4000
< 8> 4040 IF LEN(file$)>8 THEN PRINT bell$:GOTO 40
     00
<58> 4050 datfile$=file$+SPACE$(8-LEN(file$))+".LE
     X":indfile$=file$+".IND"
<87> 4060 IF FIND$(datfile$)<>"" THEN GOTO 4140 'D
     atei öffnen
< 8> 4070 '
<98> 4080 PRINT FNloc$(26,0);line$
<18> 4090 PRINT FNinfo$("Soll die Datei neu einger
     ichtet werden? <J/N>")
<29> 4100 match$="JN":GOSUB 240
<21> 4110 IF antwort=2 THEN PRINT FNloc$(26,0);loe
     sch$:GOTO 4000
<91> 4120 CREATE #1,datfile$,indfile$,2,112:CLOSE
     #1
<97> 4130 '
<78> 4140 BUFFERS 50
<28> 4150 OPEN "K",#1,datfile$,indfile$,2
<12> 4160 FIELD #1,15 AS w$,90 AS s$
< 5> 4170 RETURN
<13> 4180 '
<16> 4190 '
<90> 4200 ' *******************
<83> 4210 '... Diskettenmenue
<96> 4220 ' *******************
< 0> 4230 '
<60> 4240 PRINT FNloc$(2,0);SPC(15);"Laufwerk: ";l
     w$;SPC(30);"Datei: "+datfile$
<10> 4250 PRINT FNloc$(4,0)+invon$+"........... Di
     rectory <A><B><M> * <D>iskette <L>aufwerk <H>
     elp E<x>it <C>lear............ "+invoff$
<68> 4260 GOSUB 4500 'Directory anzeigen
< 6> 4270 match$="ABMDLHXC":GOSUB 240
<25> 4280 ON antwort GOTO 4330,4340,4350,4580,4650
     ,4810,5230
<31> 4290 IF sperre=0 THEN GOTO 410 ELSE GOTO 4270
<92> 4300 '
<95> 4310 ' ***** DIRECTORY A/B/M ******
<98> 4320 '
< 0> 4330 lwx$="A":GOTO 4360
<15> 4340 lwx$="B":GOTO 4360
<38> 4350 lwx$="M"
<83> 4360 IF lwx$=lw$ THEN PRINT bell$:GOTO 4270 '
     Menue
< 7> 4370 CLOSE #1:OPTION FILES lwx$ 'Laufwerk wec
     hseln
<91> 4380 IF lwx$="M" THEN GOTO 4420
< 8> 4390 PRINT FNloc$(26,0);line$
<34> 4400 PRINT FNinfo$("Bitte pruefen, ob eine Di
     skette eingelegt ist!")
<65> 4410 taste$=INPUT$(1)
<17> 4420 GOSUB 4500:GOSUB 310 'Directory anzeigen
      - Taste
<13> 4430 PRINT FNloesch$:PRINT FNloc$(26,0);line$
<63> 4440 PRINT FNinfo$("B I T T E.. W A R T E N !
     ")
<91> 4450 OPTION FILES lw$:lwx$=lw$ 'Laufwerk zuru
     eck
<34> 4460 IF sperre=0 THEN GOSUB 4140:GOTO 4240 EL
     SE GOTO 4250 'Datei oeffnen - Menue
<16> 4470 '
<18> 4480 '----- UP Directory anzeigen -----
<22> 4490 '
<40> 4500 PRINT FNloesch$
<86> 4510 PRINT line$:PRINT"Directory Laufwerk ";l
     wx$:PRINT line$
<19> 4520 IF FIND$("*.*")="" THEN PRINT"Es sind no
     ch keine Programme gespeichert!":GOTO 4540
<60> 4530 DIR
<46> 4540 PRINT:PRINT line$:RETURN
<12> 4550 '
```

```
<15> 4560 ' ***** DISKETTE WECHSELN *****
<18> 4570 '
<34> 4580 IF lw$="M" THEN PRINT bell$:GOTO 4270
<38> 4590 PRINT FNloc$(4,34)" Diskette "
< 9> 4600 CLOSE #1:PRINT FNloc$(2,0)+lwtext$
< 6> 4610 GOTO 120 'Datendiskette einlegen
< 5> 4620 '
< 8> 4630 ' ***** LAUFWERK WECHSELN *****
<11> 4640 '
<30> 4650 PRINT FNloc$(4,0)+invon$;SPC(30);"Laufwe
     rk <A> <B> <M> * <C>lear";SPC(30)+invoff$
<60> 4660 PRINT FNloesch$
< 4> 4670 lw$=INPUT$(1):lw$=UPPER$(lw$)
<26> 4680 IF lw$=lwx$ THEN PRINT bell$:GOTO 4670
<37> 4690 IF lw$="C" THEN lw$=lwx$:GOTO 4250
<47> 4700 IF lw$<>"A" AND lw$<>"B" AND lw$<>"M" TH
     EN GOTO 4670
<80> 4710 lwtext$=SPACE$(15)+"Laufwerk: "+lw$+SPAC
     E$(30)+"Datei: ????????.LEX"
<79> 4720 CLOSE #1:PRINT FNloc$(2,0)+lwtext$:lwx$=
     lw$
<66> 4730 IF lw$="M" THEN GOTO 150 ELSE GOTO 120 '
     Datendiskette einlegen
<13> 4740 '
<16> 4750 '
<19> 4760 ' ***** HELPTEXTE *****
<22> 4770 '
< 4> 4780 PRINT FNloc$(4,43)" Help ":GOSUB 4830:GO
     TO 490. 'Optionen-Menue
<95> 4790 PRINT FNloc$(4,57)" Help ":GOSUB 4930:GO
     TO 2100 'Printer-Menue
<77> 4800 PRINT FNloc$(4,62)" Help ":GOSUB 5020:GO
     TO 3200 'Files-Menue
<93> 4810 PRINT FNloc$(4,56)" Help ":GOSUB 5020:GO
     TO 4250 'Disketten-Menue
< 9> 4820 '
<27> 4830 PRINT FNloc$(6,0);line$
<90> 4840 PRINT FNinfo$("* * * * *. HAUPT- UND OPT
     IONENMENUE. * * * * *"):PRINT:PRINT
<45> 4850 PRINT"EINGABE.. : Eingabe eines Stichwor
     tes und des Erläuterungstextes. Jede Zeile mu
     ß mit <RE-........... TURN> beendet werden.
     Abgeschlossen wird die Eingabe mit einem Gitt
     er(#)."
<79> 4860 PRINT"SUCHEN... : Zeigt nach der Stichwo
     rteingabe einen Datensatz auf dem Monitor an.
     "
<35> 4870 PRINT"KORREKTUR : Zeigt einen Datensatz
     Zeile fuer Zeile auf dem Monitor an. Fehler k
     oennen............... durch Neueingabe der Z
     eile korrigiert werden."
<72> 4880 PRINT"LOESCHEN. : Nach Bestätigung der S
     icherheitsabfrage wird ein Datensatz gelöscht
     ."
<78> 4890 PRINT"INDEX.... : Listet das Inhaltsverz
     eichnis alphabetisch auf."
<18> 4900 PRINT"LEXIKON.. : Zeigt alle Datensätze
     der Reihe nach auf dem Monitor an.":PRINT
<76> 4910 GOTO 5130 'Exit und Clear
<11> 4920 '
<29> 4930 PRINT FNloc$(6,0);line$
< 0> 4940 PRINT FNinfo$("* * * * *. PRINTER-MENUE.
      * * * * *"):PRINT:PRINT
<89> 4950 PRINT"SUCHEN... : Druckt nach der Stichw
     orteingabe einen Datensatz aus."
<69> 4960 PRINT"INDEX.... : Druckt das Inhaltsverz
     eichnis alphabetisch aus."
<86> 4970 PRINT"LEXIKON.. : Druckt alle Datensätze
     der Reihe nach aus! Der Ausdruck kann mit <C
     AN>................. abgebrochen werden."
<25> 4980 PRINT"WAEHLEN.. : Zeigt einen Datensatz
     auf dem Monitor und druckt ihn auf Wunsch aus
     !"
<49> 4990 PRINT"ZEIGEN... : Zeigt alle Datensätze
     auf dem Monitor an und druckt einzelne auf Wu
     nsch aus!":PRINT
<56> 5000 GOTO 5130 'Exit und Clear
<90> 5010 '
<90> 5020 PRINT FNloesch$;line$
<78> 5030 PRINT FNinfo$("* * * * *. FILES- UND DIS
     KETTEN-MENUE. * * * * *"):PRINT:PRINT
< 4> 5040 PRINT"DATEIEN.. : Zeigt die Lexikon-Date
     ien des gewählten Laufwerks an!"
<93> 5050 PRINT"LOESCHEN. : Nach Bestätigung der S
     icherheitsabfrage wird die aktuelle Datei gel
     oescht!"
<38> 5060 PRINT"UMBENENNEN: Nach Eingabe des neuen
     Namens wird die aktuelle Datei umbenannt!"
<74> 5070 PRINT"WECHSELN. : Die Datei kann gewechs
     elt oder eine neue eingerichtet werden!":PRIN
     T
<12> 5080 '
<38> 5090 PRINT"DIRECTORY : Zeigt das Directory de
     s gewählten Laufwerks an!"
<36> 5100 PRINT"DISKETTE. : Die Diskette kann gewe
     chselt werden!"
<25> 5110 PRINT"LAUFWERK. : Das Laufwerk kann gewe
     chselt werden!":PRINT
<95> 5120 '
<38> 5130 PRINT"EXIT..... : Die Datei wird geschlo
     ssen, das Programm kehrt zu Basic zurueck."
<20> 5140 PRINT"CLEAR.... : Das Programm kehrt in
     das vorhergehende Menue zurueck."
<93> 5150 GOSUB 310:PRINT FNloesch$:RETURN 'Taste
< 8> 5160 '
<11> 5170 '
<14> 5180 ' ***** ENDE. *****
<17> 5190 '
<36> 5200 PRINT bell$:PRINT FNloc$(26,0);line$
<19> 5210 PRINT FNinfo$("F E H L E R !. Das Progra
     mm wird beendet!")
<47> 5220 FOR x=1 TO 2000:NEXT
<65> 5230 CLOSE #1:PRINT FNloc$(26,0);line$
< 2> 5240 PRINT FNinfo$("Die Datei wurde geschloss
     en!")
<56> 5250 FOR x=1 TO 2000:NEXT
<96> 5260 IF kor$="ein" THEN LPRINT koroff$
<40> 5270 IF mic$="ein" THEN LPRINT micoff$
<25> 5280 LPRINT esc$+"P" 'Pica-Schrift
<68> 5290 BUFFERS 0:PRINT cls$;con$:END
<93> 5300 '
<96> 5310 '
< 0> 5320 ' ***** FEHLERMELDUNGEN *****
< 3> 5330 '
<93> 5340 PRINT FNloc$(26,0);line$
<18> 5350 PRINT FNinfo$("Es wurden noch keine Date
     nsätze eingegeben!")
<79> 5360 taste$=INPUT$(1)
<77> 5370 PRINT FNloesch$:RETURN
<18> 5380 '
< 9> 5390 PRINT FNloc$(26,0);line$
<77> 5400 PRINT FNinfo$("Keinen Datensatz zu diese
     m Stichwort gefunden!")
<66> 5410 taste$=INPUT$(1)
<19> 5420 PRINT FNloc$(26,0);loesch$:RETURN
< 5> 5430 '
< 8> 5440 '
<11> 5450 ' ***** INITIALISIERUNG *****
<14> 5460 '
<42> 5470 '----- Bildschirmcodes -----
<20> 5480 '
<91> 5490 esc$=CHR$(27):cls$=esc$+"E"+esc$+"H":loe
     sch$=esc$+"J"
<31> 5500 con$=esc$+"e":coff$=esc$+"f":cup$=esc$+"
     A"
<87> 5510 invon$=esc$+"p":invoff$=esc$+"q"
< 0> 5520 bell$=CHR$(7):line$=STRING$(90,154):crlf
     $=CHR$(13)+CHR$(10)
<16> 5530 DEF FNloc$(x,y)=esc$+"Y"+CHR$(32+x)+CHR$
     (32+y)
<77> 5540 DEF FNloesch$=esc$+"Y"+CHR$(32+6)+CHR$(3
     2+0)+loesch$
<22> 5550 DEF FNinfo$(text$)=STRING$((80-LEN(text$
     ))/2,32)+invon$+"... "+text$+"... "+invoff$+S
     TRING$((80-LEN(text$))/2,32)+crlf$+line$
<37> 5560 lw$="A":lwx$=lw$:nor$="ein":kor$="aus":m
     ic$="aus"
<95> 5570 lwtext$=SPACE$(15)+"Laufwerk: "+lw$+SPAC
     E$(30)+"Datei: ????????.LEX"
<58> 5580 index=0:suchen=0:lexikon=0:sperre=0
<25> 5590 '
<55> 5600 '----- Druckercodes -----
< 3> 5610 '
<87> 5620 elite$=esc$+"M":dop$=CHR$(14):strich$=ST
     RING$(90,45)
< 8> 5630 koron$=esc$+"m"+"1":koroff$=esc$+"m"+"0"
<47> 5640 micon$=esc$+"S"+CHR$(0)+esc$+"SI"+CHR$(1
     5)+esc$+"3"+CHR$(20)
<59> 5650 micoff$=esc$+"T"+esc$+"DC2"+CHR$(18)+esc
     $+"2"+elite$
< 5> 5660 druck=0:worte=0:ausdruck=0
<54> 5670 LPRINT elite$
<24> 5680 '
<38> 5690 '----- Titel ------
< 2> 5700 '
<79> 5710 PRINT cls$;coff$
<94> 5720 PRINT line$:PRINT:PRINT line$:PRINT:PRIN
     T line$
<53> 5730 PRINT FNloc$(2,0)+lwtext$
<61> 5740 PRINT FNloc$(4,0)+invon$+"... * * * * *
      * * * * * * * * *.. L E X I K O N . B A S.. *
      * * * * * * * * * * * * *.... "+invoff$
<12> 5750 RETURN
```

Bounce

100 Level Spielvergnügen

BOUNCE!! Ist es nicht das, was Sie schon immer für Ihren PCW gesucht haben? Ein Spiel, das Sie nicht mehr losläßt, sobald Sie es einmal begonnen haben? Nun, vor Ihnen liegt es. BOUNCE erfordert logische Vorgehensweise, stellt Anforderungen an Ihre Geschicklichkeit und verlangt zudem starke Nerven. Das Programm zeichnet sich durch Pulldown-Menüs, akustische und grafische Effekte aus, aber sehen Sie selbst und lassen Sie sich überraschen:

Auf einem acht x acht grossen Feld befinden sich Spielsteine, deren Position einer bestimmten Gesetzmässigkeit unterliegen. Diese sind von einem vorgegebenen Startstein beginnend durch Springen auf einen anderen Stein zu vernichten. Dabei gibt es acht verschiedene Sprungmöglichkeiten: Mit den vier Cursor-Tasten alleine können einfache Sprünge, in Verbindung mit der SHIFT-Taste jedoch Doppelsprünge in horizontaler und vertikaler Richtung ausgeführt werden. Der jeweils verlassene Stein wird vernichtet. Ihre Aufgabe ist es, in hundert Sekunden das gesamte Spielfeld abzuräumen. Das Limit wird im Info-Fenster und als schrumpfender Balken links neben dem Spielfeld angezeigt. Sollte das Limit bis auf dreißig Sekunden ablaufen, so fängt die gesamte Bildschirmdarstellung an zu wackeln. Verstreicht das Limit oder haben Sie sich versprungen, so ist das Spiel verloren.

Die auf Diskette vorhandenen Level werden nach einem Zufallssystem ausgesucht, jedoch so, daß sich die Level erst dann wiederholen, wenn alle vorhandenen gespielt worden sind. In diesem Fall jedoch werden diese in anderer Form als zuvor dargestellt (andere Startposition (2), spiegelverkehrt (2), gedreht (4)). Die Zahlen in Klammern geben die Anzahl der Möglichkeiten an. Insgesamt kann also jeder Level in 2x2x4=16 Variationen auftreten.

Während des Spiels wandeln sich gelegentlich normale Spielsteine in die Buchstaben b,o,u,n,c,e um, die für zehn Sekunden stehen bleiben. Gelingt es Ihnen, in dieser Zeit den aufgetauchten Buchstaben zu vernichten, so gibt es Extrapunkte, andernfalls wandelt sich der Buchstabe wieder in einen normalen Spielstein um.

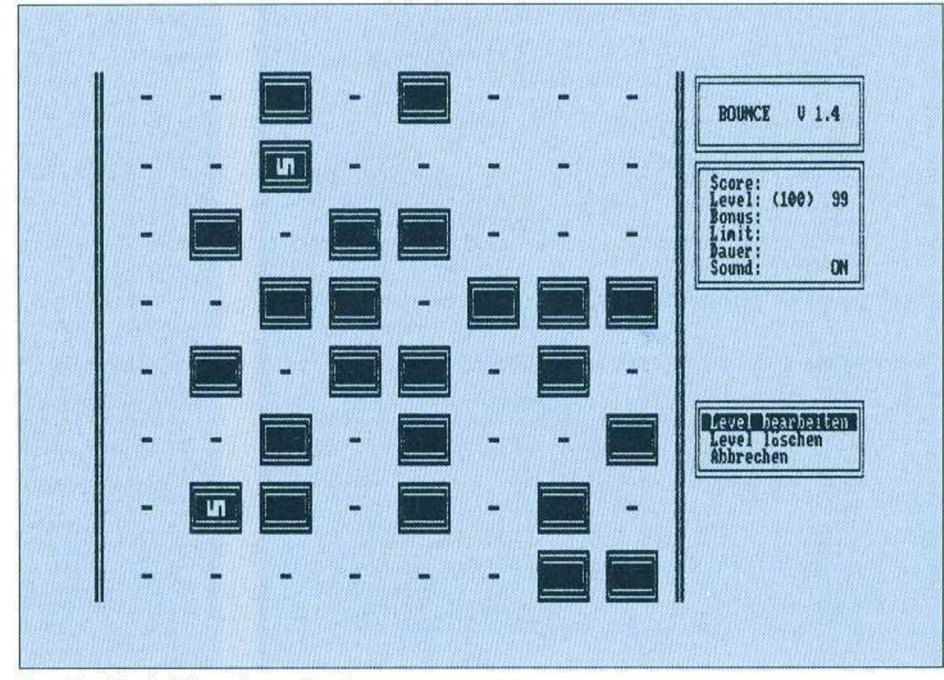
Sollten Sie einmal alle sechs verschiedenen Buchstaben vernichtet haben, so daß das Wort "bounce" komplett ist, so gibt es nochmals Extrapunkte. Die Anzahl der durch Sie vernichteten Buchstaben wird im Infofenster angezeigt. Mit STOP können Sie während des Spiels eine Pause einlegen. Dabei wird das Spielfeld gelöscht, die Pause angezeigt und das Spiellimit angehalten. Bereits erschienene, nicht vernichtete Buchstaben gehen verloren. Wollen Sie das Spiel fortsetzen, so drücken Sie eine beliebige Taste. **Das Drücken der PTR-Taste während des Spiels hat keinen Einfluss auf das Limit!**
Sollten Sie bis spät in die Nacht spielen und möchten dabei niemand stören, so können Sie mit AUSBL alle akustischen Signale aus- und auch wieder einschalten.
Wenn Ihnen während des Spiels ein Level auffallen sollte, den Sie vielleicht anschließend bearbeiten möchten, so drücken Sie COPY, damit sich BOUNCE diesen Level merkt. Wie unten beschrieben, können Sie diesen dann im Editor durch erneutes Drücken von COPY aufrufen.

Die Highscore-Liste

BOUNCE verfügt über eine Highscore-Liste, in die Sie BOUNCE automatisch eintragen läßt, wenn Sie mit Ihren erspielten Punkten einen der ersten zehn Plätze belegen. Diese Highscore-Liste wird auf Diskette abgespeichert. Existiert diese bei Programmstart nicht auf Diskette, so wird eine Liste erstellt, in der alle zehn Plätze mit zu überbietenden Punkten belegt sind. Die aktuelle Highscore-Liste können Sie sich jederzeit im Hauptmenü zeigen lassen. Abgespeichert werden dabei Ihr Name, Ihre erzielten Punkte, der erreichte Level, die Anzahl der durch Sie vernichteten Buchstaben und die gesamte Spieldauer in Sekunden. Bei Punktgleichheit entscheidet die Anzahl gespielter Level, dann die Anzahl vernichteter Buchstaben, dann die kürzere Spieldauer und letztendlich das neuere Ergebnis über die bessere Platzierung.

Der Editor

BOUNCE enthält einen Editor, mit dem Sie eigene Level erstellen und auf Diskette abspeichern können. Wenn Sie ihn im Hauptmenü aufrufen, so wird Ihnen im Info-Fenster in Klammern angezeigt, wieviele Level bereits existieren. Sie werden nun gefragt, ob Sie einen Level aufrufen oder erstellen wollen. Im ersten Fall springt der Cursor ins Infofenster und erwartet die Eingabe



Erstellen Sie sich Ihre eigenen Level

des gewünschten Levels. Drücken Sie ENTER, ohne eine Eingabe gemacht zu haben, so ruft BOUNCE den zuletzt aufgerufenen beziehungsweise. bearbeiteten auf. Wenn Sie COPY drücken, ruft BOUNCE automatisch den durch Sie während des Spiels markierten Level auf. Haben Sie jedoch zuvor keinen Level aufgerufen oder markiert oder haben Sie einen Level gelöscht, so wird der erste Level aufgerufen.

Durch Drücken der Taste zwischen den Cursor-Tasten setzen Sie einen Stein oder löschen einen bereits gesetzten. Mit FORM bestimmen Sie eine Startposition oder löschen eine bereits bestimmte. Pro Level müssen zwei Startpositionen angegeben werden.

Wenn Sie einen Level ordnungsgemäß mit zwei Startpositionen erstellt haben, so drücken Sie ENTER, damit BOUNCE Sie fragen kann, ob der Level in dieser Form abgespeichert werden kann. Das Programm schließt nach einem Schreibzugriff die betroffene Datei und öffnet sie wieder, damit zu jeder Zeit die Daten gesichert sind. Mit CAN brechen Sie fast jede gewählte Operation ab.

Zum Listing BOUNCE.BAS: Um das Programm BOUNCE lauffähig zu machen, ist die Tastaturbelegung geringfügig zu ändern: Hierfür erstellen Sie mit dem Programm RPED.BAS die Datei BOUNCE.WP

07 N "↑'251'" <RETURN>
14 S "↑'252'" <RETURN>
79 S "↑'253'" <RETURN>
15 S "↑'254'" <RETURN>
06 S "↑'255'" <RETURN>

Diese Änderung ist notwendig, um SHIFT-Cursor - Kombinationen und die Benutzung der Taste zwischen den Cursor-Tasten zu ermöglichen. Vor dem Programmstart müssen Sie dann unter CP/M mit dem Programm SETKEYS.COM (Seite zwei der Systemdisketten) die Tastatur mit der Eingabe "SETKEYS BOUNCE.WP" umdefinieren und das Spiel mit "BASIC BOUNCE" starten.

Zum Listing LEVEL.BAS: Dieses Listing enthält hundert ausgeklügelte und getestete Level für BOUNCE, das nach dem Programmstart die Datei BOUNCE.LEV eröffnet und alle Level spielfertig einrichtet. Jeweils fünf DATA-Werte ergeben einen Level.

Hannes Geyer/rs

In letzter Minute konnte noch ein Fehler im Programm ausfindig gemacht werden, der sich auf die Berechnung der Extrabounspunkte auswirkt. Um diesen Fehler zu korrigieren, ändern Sie einfach die folgenden drei Zeilen wie angegeben ab:

```
1130 ge=1:bw=INT(6*RND):
s=8*xw-4:FOR z=4*yw-4
TO 4*yw-2:GOSUB 90:PRINT
SPC(6):NEXT z:z=4*yw-1:FOR
i=1 TO 3:z=z-1:GOSUB 90;
RINT inv$;MID$(bou$,8*bw+1+
(z-4*yw+4)*48,6 );nrm$:NEXT i:
zv=se
1370 ge=0:bon=bon+1:GOSUB
2670:GOSUB 2960:pun=pun+
(2*lim MOD 10):ep1=INT(2*lim/
10):ep2=10:gosub 2870:IF
(gbs AND 2↑bw)=0 THEN gbs=
gbs+2↑bw
1385 GOSUB 3020
```

Auf der DATABOX steht selbstverständlich die korrigierte Fassung von BOUNCE zur Verfügung.

```
LISTING >BOUNCE  <, REMARK = >'<.

<67> 10 '****************************************
     *****
<22> 20 '*............. BOUNCE.. V 1.4..........
     ... *
<36> 30 '* (C) 12.05.1991 by Hannes Geyer, 4300 Es
     sen *
<44> 40 '*.... & CPC International..............
     ... *
<97> 50 '*... für PCW 8256/8512 - Schneider JOYCE.
     ... *
<64> 60 '*.. Mallard BASIC...... 18 Total 1k Block
     s.. *
<73> 70 '****************************************
     *****
<27> 80 OPTION RUN:CLEAR ,,,2,35
<51> 90 DEFINT a-o,q-z:WIDTH 255:GOTO 110
<72> 100 PRINT esc$;"Y";CHR$(z+32);CHR$(s+32);:RET
     URN
< 2> 110 DIM men$(6,4),ste$(3),jn(25),mo(6),fe(8,8
     ),v(6),b1(15),b2(15),zm(16),mt(16),bex(2),bey
     (2)
<48> 120 esc$=CHR$(27):clr$=esc$+"E"+esc$+"H"
<25> 130 inv$=esc$+"p":nrm$=esc$+"q"
<43> 140 uns$=esc$+"f":sic$=esc$+"e"
<97> 150 und$=esc$+"r":nun$=esc$+"u"
<84> 160 he$=esc$+"b"+CHR$(63)+esc$+"c"+CHR$(0)
<98> 170 du$=esc$+"b"+CHR$(0)+esc$+"c"+CHR$(63)
<60> 180 so$=" ON":so1$="OFF":lem=1:lds=1:tw=11:tw
     1=12:w1=255:w2=0:u1=255:u2=0
<90> 190 men$(1,1)=" BOUNCE starten.. ":men$(1,2)=
     " Highscore-Liste. ":men$(1,3)=" Editor aufru
     fen. ":men$(1,4)=" BOUNCE Ende..... "
<70> 200 men$(2,1)=" Level erstellen. ":men$(2,2)=
     " Level aufrufen.. ":men$(2,3)=" Editieren En
     de.. "
< 4> 210 men$(3,1)=" Level speichern. ":men$(3,2)=
     " Level unkorrekt. ":men$(3,3)=" Abbrechen...
     .... "
<67> 220 men$(4,1)=" Level bearbeiten ":men$(4,2)=
     " Level löschen... ":men$(4,3)=men$(3,3)
<89> 230 men$(5,1)=" Bestätigen...... ":men$(5,2)=
     men$(3,3)
<12> 240 men$(6,1)=" -> Mallard BASIC ":men$(6,2)=
     " -> CP/M......... ":men$(6,3)=" -> Modus 24
     x 80 ":men$(6,4)=" -> Systemstart.. "
```

```
<15> 250 lo$=CHR$(134):ro$=CHR$(140):lu$=CHR$(131)
     :ru$=CHR$(137):ve$=CHR$(133):ho$=STRING$(18,1
     38)
<56> 260 ste$(1)=lo$+STRING$(4,138)+ro$:ste$(2)=ve
     $+"... "+ve$:ste$(3)=lu$+STRING$(4,138)+ru$
<41> 270 a$=STRING$(4,154):b$=CHR$(149)+"... "+CHR
     $(149)+". ":c$=CHR$(150)+a$+CHR$(156)+". ":d$
     =CHR$(147)+a$+CHR$(153)+". "
<24> 280 bou$=CHR$(148)+SPACE$(7)+c$+CHR$(148)+"..
     . "+CHR$(148)+". "+c$+c$+c$+CHR$(151)+RIGHT$(
     c$,7)+b$+b$+b$+CHR$(149)+SPACE$(7)+CHR$(151)+
     RIGHT$(d$,7)+d$+d$+d$+CHR$(145)+"... "+CHR$(1
     45)+". "+d$+LEFT$(d$,5)+CHR$(152)+". "
<11> 290 FOR i=1 TO 6:mo(i)=i:NEXT i
<47> 300 pa$=inv$+". "+nrm$:pa$(1)=pa$+pa$+pa$:pa$
     (2)=pa$+". "+pa$:pa$(3)=pa$:pa$(4)="... "+pa$
<25> 310 cue$=" "+CHR$(149)+CHR$(146)+CHR$(152)+CH
     R$(149)+" "
<55> 320 FOR i=1 TO 8:eli$=eli$+".. "+CHR$(146)+CH
     R$(152)+".. ":NEXT i
<75> 330 nor$=esc$+"X. ?y":lf1$=esc$+"X #> "+clr$+
     nor$:lf2$=esc$+"X-f13"+clr$+nor$:lf3$=esc$+"X
     &n$*"+clr$+nor$:lf4$=esc$+"X. > "+clr$+nor$
<42> 340 ba=16:FOR i=1 TO 15:READ b1(i),b2(i):NEXT
      i
<85> 350 FOR i=1 TO 25:READ jn(i):NEXT i
<44> 360 DATA 128,144,145,148,146,152,139,142,149,
     133,211,47,135,141,150,153,60,62,138,154,138,
     32,149,32,131,140,220,221,240,244,1,2,1,3,3,1
     ,2,1,2,2,2,2,2,2,1,1,3,1,4,1,1,3,1,3,1
<10> 370 RANDOMIZE (PEEK(&HFBF7)+1)*(PEEK(&HFBF8)+
     1)
<92> 380 PRINT du$;uns$;nrm$;nor$;clr$;
<38> 390 '
< 6> 400 '***** Bildschirmaufbau *****************
     *****
<23> 410 '
<43> 420 OUT 248,8:OUT 245,91:OUT 246,115:GOSUB 29
     60
<78> 430 h1=141:h2=115:OUT 248,7:FOR i=1 TO 114:h1
     =h1+1:h2=h2-1:OUT 246,h1:FOR j=1 TO 30:NEXT j
     :OUT 246,h2:FOR j=1 TO 30:NEXT j:IF INKEY$=""
      THEN NEXT i ELSE ub=1
<54> 440 OUT 246,0:r1=SGN(0.5-INT(2*RND)):r2=SGN(0
     .5-INT(2*RND)):FOR i=0 TO 30:z=r1*(i-15)+15:s
     =1:GOSUB 100:PRINT inv$;CHR$(149):z=r2*(i-15)
     +15:s=68:GOSUB 100:PRINT CHR$(149);nrm$:NEXT
     i
```

```
<42> 450 z=0:s=70:GOSUB 100:PRINT lo$;ho$;ro$
<14> 460 FOR z=1 TO 3:GOSUB 100:PRINT ve$;SPC(18);
     ve$:NEXT z
<17> 470 z=4:GOSUB 100:PRINT lu$;ho$;ru$:z=5:GOSUB
      100:PRINT lo$;ho$;ro$
<90> 480 FOR z=6 TO 11:GOSUB 100:PRINT ve$;SPC(18)
     ;ve$:NEXT z
<69> 490 z=12:GOSUB 100:PRINT lu$;ho$;ru$
<22> 500 z=2:s=73:GOSUB 100:PRINT "BOUNCE.. V 1.4"
     :z=6:s=72:GOSUB 100:PRINT "Score:":z=7:GOSUB
     100:PRINT "Level:":z=8:GOSUB 100:PRINT "Bonus
     :":z=9:GOSUB 100:PRINT "Limit:":z=10:GOSUB 10
     0:PRINT "Dauer:":z=11:GOSUB 100:PRINT "Sound:
     ....... ON"
<80> 510 IF ub=1 THEN 600
<76> 520 FOR u=1 TO 4:FOR i=6 TO 1 STEP -1:SWAP mo
     (INT(i*RND)+1),mo(i):f=mo(i)
<76> 530 s=8*f+4:FOR z=14 TO 16:GOSUB 100:PRINT SP
     C(6):NEXT z:FOR z=16 TO 14 STEP -1:GOSUB 100
<41> 540 IF u=1 OR u=3 THEN PRINT inv$;ste$(z-13);
     nrm$:GOTO 570
<28> 550 IF u=2 THEN PRINT inv$;
<50> 560 PRINT MID$(bou$,8*f-7+(z-14)*48,6);nrm$
<79> 570 IF INKEY$>"" THEN GOSUB 2960:GOTO 600
<26> 580 NEXT z:FOR j=1 TO 80:NEXT j,i:FOR j=1 TO
     1000:NEXT j,u:GOTO 520
<40> 590 '
< 2> 600 IF FIND$("BOUNCE.SCO")="" THEN ng=1
<76> 610 GOSUB 2600:GOSUB 2630:sw=1024:sw1=sw
<82> 620 IF sw<1 THEN 670
<62> 630 sw1=sw1/2:GET 1,sw:a1=VAL(MID$(stu$,17,2)
     ):de=EOF(1):IF sw>999 AND a1>0 THEN GOSUB 302
     0:GOTO 3050
<69> 640 GET 1,sw+1:a2=VAL(MID$(stu$,17,2))
<58> 650 IF de<0 OR a1=0 THEN sw=sw-sw1:GOTO 620
<20> 660 IF de=0 AND a2>0 THEN sw=sw+sw1:GOTO 620
<68> 670 IF ng=1 THEN FOR i=1 TO 10:na$(i)="BOUNCE
     "+SPACE$(10):pu(i)=100*(11-i):le(i)=INT((10-i
     )/3)+1:bo(i)=le(i)-1:pt(i)=20*le(i):NEXT i:GO
     SUB 2620:GOTO 720
<98> 680 FOR i=1 TO 10:GET 2,i:na$(i)=name$:pu(i)=
     VAL(punkte$):le(i)=VAL(level$):bo(i)=VAL(bonu
     s$):pt(i)=VAL(dauer$):NEXT i
<41> 690 '
<16> 700 '***** Hauptprogramm *********************
     *****
<26> 710 '
<10> 720 DIM zuf(sw),moe(sw,2):FOR i=1 TO sw:zuf(i
     )=i:NEXT i
<43> 730 k=1:k1=4:GOSUB 2720:PRINT lf1$:IF v(1)<>2
      THEN PRINT lf4$;lf3$
<87> 740 IF v(1)=1 THEN GOSUB 3000:st=0:ge=0:pun=0
     :lev=0:bon=0:pti=0:gbs=0:GOTO 780
<41> 750 IF v(1)=2 THEN GOSUB 2510:GOTO 730
<49> 760 IF v(1)=3 THEN swa=sw:GOSUB 2660:GOSUB 30
     00:GOTO 1750 ELSE 3050
<38> 770 '
<45> 780 FOR zu=sw TO 1 STEP -1
<28> 790 PRINT lf4$;inv$;und$;:s=0:FOR z=30 TO 0 S
     TEP -1:GOSUB 100:PRINT " ":NEXT z:PRINT nun$;
     nrm$
<42> 800 u=0:r=0:gc=FRE(""):lev=lev+1:lim=100:GOSU
     B 2650:GOSUB 2670:GOSUB 2680:GOSUB 2690:GOSUB
      2700
<60> 810 SWAP zuf(INT(zu*RND)+1),zuf(zu)
<97> 820 n=zuf(zu):GET 1,n:GOSUB 2860
<43> 830 FOR q=1 TO 2:FOR a=0 TO 7:r=r+1
<98> 840 IF (moe(n,q) AND 2^a)=0 THEN zm(r)=0:u=u+
     1:mt(u)=r ELSE zm(r)=1
< 1> 850 NEXT a,q:m=mt(INT(u*RND)+1):zm(m)=1
<31> 860 IF m>8 THEN m=m-8:d=2 ELSE d=1
<42> 870 moe(n,d)=0:FOR a=0 TO 7
<36> 880 IF zm(a+d*8-7)=1 THEN moe(n,d)=moe(n,d)+2
     ^a
<30> 890 NEXT a
<66> 900 IF moe(n,1)+moe(n,2)=510 THEN moe(n,1)=0:
     moe(n,2)=0
<28> 910 '
<85> 920 FOR y=1 TO 8:by=VAL("&H"+MID$(stu$,2*y-1,
     2))
<44> 930 FOR a=0 TO 7:IF (by AND 2^a)=0 THEN 1050
<43> 940 x=a+1:ON m GOTO 950,960,970,980,990,1000,
     1010,1020
<87> 950 x1=x:y1=y:GOTO 1030
<22> 960 x1=y:y1=9-x:GOTO 1030
<87> 970 x1=9-x:y1=9-y:GOTO 1030
<39> 980 x1=9-y:y1=x:GOTO 1030
<35> 990 x1=x:y1=9-y:GOTO 1030
<63> 1000 x1=y:y1=x:GOTO 1030
<25> 1010 x1=9-x:y1=y:GOTO 1030
```

```
<69> 1020 x1=9-y:y1=9-x:GOTO 1030
<38> 1030 fe(x1,y1)=1:st=st+1:IF a=bex(d)-1 AND y=
     bey(d) THEN ax=x1:ay=y1
<75> 1040 z=4*y1-4:s=8*x1-4:GOSUB 100:PRINT inv$;s
     te$(1):z=z+1:GOSUB 100:PRINT ste$(2):z=z+1:GO
     SUB 100:PRINT ste$(3);nrm$
<21> 1050 NEXT a,y
<82> 1060 GOSUB 3030:ptio=pti:ba=ba-1:IF ba<1 THEN
      ba=15
<69> 1070 f=INT(ba*RND)+1:SWAP b1(f),b1(ba):SWAP b
     2(f),b2(ba):POKE &HFBF7,0:POKE &HFBF8,0
< 8> 1080 '
<87> 1090 ta$=INKEY$:IF ta$>"" THEN 1230
<35> 1100 SWAP b1(ba),b2(ba):z=4*ay-3:s=8*ax-2:GOS
     UB 100:PRINT CHR$(b1(ba));CHR$(b2(ba))
<98> 1110 IF lim<30 THEN OUT 246,u1:SWAP u1,u2:hr=
     253+INT(lim/10):IF u1<u2 THEN u2=hr ELSE u1=h
     r
<49> 1120 IF RND>0.01 OR ge=1 THEN 1150
<74> 1130 xw=INT(8*RND)+1:yw=INT(8*RND)+1:IF fe(xw
     ,yw)=0 OR (xw-ax)^2+(yw-ay)^2<2 THEN 1170
<37> 1140 ge=1:bw=INT(6*RND):s=8*xw-4:FOR z=4*yw-4
      TO 4*yw-2:GOSUB 100:PRINT SPC(6):NEXT z:z=4*
     yw-1:FOR i=1 TO 3:z=z-1:GOSUB 100:PRINT inv$;
     MID$(bou$,8*bw+1+(z-4*yw+4)*48,6);nrm$:NEXT i
     :zv=se:IF (gbs AND 2^bw)=0 THEN gbs=gbs+2^bw
<77> 1150 ta$=INKEY$:IF ta$>"" THEN 1230
<64> 1160 IF ge=1 THEN IF se>zv+10 THEN ge=0:s=8*x
     w-4:FOR z=4*yw-4 TO 4*yw-2:GOSUB 100:PRINT SP
     C(6):NEXT z:FOR z=4*yw-2 TO 4*yw-4 STEP -1:GO
     SUB 100:PRINT inv$;ste$(z-4*yw+5);nrm$:NEXT z
<43> 1170 se=VAL(HEX$(PEEK(&HFBF7)))*60+VAL(HEX$(P
     EEK(&HFBF8))):IF se>100 THEN se=100
<52> 1180 lim=100-se:pti=ptio+se:GOSUB 2690:GOSUB
     2700
<89> 1190 ta$=INKEY$:IF ta$>"" THEN 1230
<63> 1200 PRINT esc$;"X. ";CHR$(31+se/3.27);" ";cl
     r$;nor$
< 7> 1210 IF lim>0 THEN 1090 ELSE 1460
<93> 1220 '
<96> 1230 ta=ASC(ta$):ax1=ax:ay1=ay
<38> 1240 IF ta=1 THEN ax=ax-1:GOTO 1360
<53> 1250 IF ta=6 THEN ax=ax+1:GOTO 1360
<70> 1260 IF ta=30 THEN ay=ay+1:GOTO 1360
<37> 1270 IF ta=31 THEN ay=ay-1:GOTO 1360
<93> 1280 IF ta=252 THEN ay=ay-2:GOTO 1360
<58> 1290 IF ta=253 THEN ay=ay+2:GOTO 1360
<52> 1300 IF ta=254 THEN ax=ax-2:GOTO 1360
<17> 1310 IF ta=255 THEN ax=ax+2:GOTO 1360
<16> 1320 IF ta=3 THEN ge=0:GOSUB 2900
<21> 1330 IF ta=21 THEN GOSUB 2980
<85> 1340 IF ta=23 THEN lem=n
<31> 1350 GOTO 1090
<82> 1360 OUT 248,tw:st=st-1:fe(ax1,ay1)=0:OUT 248
     ,12:PRINT esc$;"X";CHR$(4*ay1+28);CHR$(8*ax1+
     28);CHR$(34);CHR$(37);clr$;nor$
<47> 1370 IF ge=0 OR ax1<>xw OR ay1<>yw THEN 1400
<84> 1380 ge=0:bon=bon+1:GOSUB 2680:GOSUB 2970:pun
     =pun+(2*lim MOD 10):ep1=INT(2*lim/10):ep2=10:
     GOSUB 2880:GOSUB 3030
<70> 1390 IF gbs>62 THEN gbs=0:FOR i=1 TO 3:PRINT
     lf2$;he$:FOR j=1 TO 200:NEXT j:PRINT du$:FOR
     bw=0 TO 5:GOSUB 2970:NEXT bw,i:ep1=25-INT(0.2
     5*pti/lev):ep2=100:GOSUB 2880:PRINT lf2$
<88> 1400 pun=pun+10:GOSUB 2650:IF st>0 THEN 1440
<66> 1410 FOR i=1 TO 3:FOR j=1 TO 200:NEXT j:OUT 2
     48,tw:GOSUB 100:OUT 248,12:NEXT i:FOR j=1 TO
     200:NEXT j:pun=pun+(lim MOD 5):ep1=INT(lim/5)
     :ep2=5:GOSUB 2880
< 9> 1420 SWAP w1,w2:FOR i=w1 TO w2 STEP SGN(1-w1)
     :OUT 246,i:FOR j=1 TO 1:NEXT j,i:OUT 246,0
<60> 1430 NEXT zu:GOTO 780
<55> 1440 IF ax<1 OR ax>8 OR ay<1 OR ay>8 THEN 146
     0
<54> 1450 IF fe(ax,ay)>0 THEN 1090
<93> 1460 OUT 246,0:pti=ptio+se:GOSUB 2700
<67> 1470 FOR i=1 TO 10:FOR j=200 TO 220:OUT 248,t
     w:OUT 248,12:OUT 245,j:NEXT j,i:OUT 245,91
<56> 1480 s=70:FOR z=26 TO 30:GOSUB 100:PRINT inv$
     ;SPC(20):NEXT z:z=28:s=76:GOSUB 100:PRINT "GA
     ME OVER";nrm$
<19> 1490 '
<81> 1500 '***** Highscore ************************
     *****
<96> 1510 '
<58> 1520 FOR h=10 TO 1 STEP -1
<73> 1530 IF pun>pu(h) THEN 1590
<41> 1540 IF pun<pu(h) THEN 1600
<91> 1550 IF lev>le(h) THEN 1590
<59> 1560 IF lev<le(h) THEN 1600
```

```
<29> 1570 IF bon>bo(h) THEN 1590
<82> 1580 IF bon<bo(h) OR pti>pt(h) THEN 1600
<98> 1590 NEXT h
<82> 1600 IF h=10 THEN 730
<31> 1610 FOR j=10 TO h+2 STEP -1:na$(j)=na$(j-1):
     pu(j)=pu(j-1):le(j)=le(j-1):bo(j)=bo(j-1):pt(
     j)=pt(j-1):NEXT j
< 7> 1620 na$(h+1)=SPACE$(16):pu(h+1)=pun:le(h+1)=
     lev:bo(h+1)=bon:pt(h+1)=pti
<45> 1630 PRINT lf1$:GOSUB 2510:na$(h+1)="":GOSUB
     3030:z=2*h+8:s=15:GOSUB 100:PRINT sic$;
<43> 1640 ta$=INPUT$(1):ta=ASC(ta$):IF ta>32 AND t
     a<127 THEN 1700
< 3> 1650 IF ta=32 THEN IF s=15 THEN 1640 ELSE 170
     0
<62> 1660 IF ta=127 THEN IF s>15 THEN na$(h+1)=LEF
     T$(na$(h+1),LEN(na$(h+1))-1):s=s-1:GOSUB 100:
     PRINT " ";:GOSUB 100 ELSE GOSUB 3020
< 1> 1670 IF ta=13 THEN IF na$(h+1)="" THEN GOSUB
     3020:GOTO 1640 ELSE na$(h+1)=na$(h+1)+SPACE$(
     31-s):PRINT uns$:GOSUB 2620:GOTO 730
<42> 1680 IF ta=21 THEN GOSUB 2980
<56> 1690 GOTO 1640
< 5> 1700 IF s<31 THEN GOSUB 100:PRINT ta$;:na$(h+
     1)=na$(h+1)+ta$:s=s+1:GOSUB 100 ELSE GOSUB 30
     20
<34> 1710 GOTO 1640
< 4> 1720 '
<13> 1730 '***** Editor *************************
     *****
<10> 1740 '
<86> 1750 k=2:k1=3:GOSUB 2720
<42> 1760 IF v(2)=3 OR ta=8 THEN PRINT lf3$:IF sw<
     >swa THEN ERASE zuf,moe:GOTO 720 ELSE 730
<61> 1770 IF v(2)=1 THEN d=0:lev=sw+1:GOSUB 2670:G
     OSUB 3010:GOTO 2050
<91> 1780 num$="":z=7:s=85:GOSUB 100:PRINT sic$;
<12> 1790 ta=ASC(INPUT$(1))
< 6> 1800 IF ta=8 THEN PRINT uns$:GOSUB 2660:GOTO
     1750
<59> 1810 IF ta=13 THEN 1870
<12> 1820 IF ta=127 THEN IF s=85 THEN GOSUB 3020 E
     LSE num$=LEFT$(num$,LEN(num$)-1):s=s-1:GOSUB
     100:PRINT " ";:GOSUB 100
<31> 1830 IF ta=21 THEN GOSUB 2980
<84> 1840 IF ta=23 THEN num$=STR$(lem):GOTO 1870
<79> 1850 IF ta>47 AND ta<58 THEN IF s=88 THEN GOS
     UB 3020 ELSE num$=num$+CHR$(ta):GOSUB 100:PRI
     NT CHR$(ta);:s=s+1:GOSUB 100
<29> 1860 GOTO 1790
<41> 1870 PRINT uns$;:lev=VAL(num$):IF lev<1 OR le
     v>sw THEN lev=lds
<27> 1880 GOSUB 2670:GET 1,lev:lds=lev:FOR y=1 TO
     8:by=VAL("&H"+MID$(stu$,2*y-1,2)):FOR a=0 TO
     7:b=2^a
<55> 1890 IF (by AND b)<>0 THEN fe(a+1,y)=1
<65> 1900 NEXT a,y:GOSUB 2860:GOSUB 3010:FOR b=1 T
     O 8:FOR a=1 TO 8
<18> 1910 IF fe(a,b)>0 THEN z=4*b-4:s=8*a-4:GOSUB
     100:PRINT inv$;ste$(1):z=z+1:GOSUB 100:PRINT
     ste$(2):z=z+1:GOSUB 100:PRINT ste$(3);nrm$
<37> 1920 NEXT a,b:d=2:FOR i=1 TO 2:fe(bex(i),bey(
     i))=2:z=4*bey(i)-3:s=8*bex(i)-2:GOSUB 100:PRI
     NT CHR$(145);CHR$(148):NEXT i
<11> 1930 '
<31> 1940 '***** Bearbeiten / Löschen ************
     *****
<17> 1950 '
<10> 1960 k=4:k1=3:GOSUB 2720
<80> 1970 IF v(4)=3 OR ta=8 THEN GOSUB 2660:PRINT
     lf1$:GOSUB 3000:GOTO 1750
<53> 1980 IF v(4)=1 THEN x=bex(1):y=bey(1):z=4*y-3
     :s=8*x-3:GOSUB 100:PRINT inv$;CHR$(149):s=s+3
     :GOSUB 100:PRINT CHR$(149);nrm$:s=s-4:GOTO 20
     60
<14> 1990 k=5:k1=2:GOSUB 2720
<38> 2000 IF v(5)=2 OR ta=8 THEN 1960
<42> 2010 GET 1,sw:PUT 1,lev:LSET stu$="":PUT 1,sw
     :sw=sw-1:lem=1:lds=1:GOTO 2470
<90> 2020 '
<48> 2030 '***** Level erstellen *****************
     *****
<96> 2040 '
<36> 2050 x=1:y=1:z=1:s=4:GOSUB 100:PRINT cue$
<77> 2060 afe=fe(x,y)
<89> 2070 ta=ASC(INPUT$(1)):IF ta<>1 AND ta<>6 AND
      ta<>8 AND ta<>13 AND ta<>18 AND ta<>21 AND t
     a<>30 AND ta<>31 AND ta<251 THEN 2070
<65> 2080 IF ta=1 THEN x=x-1:IF x=0 THEN x=8
<81> 2090 IF ta=6 THEN x=x+1:IF x=9 THEN x=1
< 7> 2100 IF ta=30 THEN y=y+1:IF y=9 THEN y=1
<70> 2110 IF ta=31 THEN y=y-1:IF y=0 THEN y=8
<66> 2120 IF ta=13 THEN IF d<>2 THEN GOSUB 3020:GO
     TO 2070
<43> 2130 IF ta=18 THEN 2330
<10> 2140 IF ta=8 THEN GOSUB 2660:PRINT lf1$:GOSUB
      3000:GOTO 1750
<17> 2150 IF ta=21 THEN GOSUB 2980:GOTO 2070
<21> 2160 IF ta=251 THEN 2300
<13> 2170 IF ta=252 THEN y=y-2:IF y<1 THEN y=8
<80> 2180 IF ta=253 THEN y=y+2:IF y>8 THEN y=1
<33> 2190 IF ta=254 THEN x=x-2:IF x<1 THEN x=8
<72> 2200 IF ta=255 THEN x=x+2:IF x>8 THEN x=1
<13> 2210 GOSUB 100
<29> 2220 IF afe=0 THEN PRINT ". ";CHR$(146);CHR$(
     152);" "
<86> 2230 IF afe>0 THEN s=s+1:GOSUB 100:PRINT inv$
     ;" ":s=s+3:GOSUB 100:PRINT " ";nrm$:s=s-4
<16> 2240 IF ta=13 THEN 2390
<69> 2250 z=4*y-3:s=8*x-4:GOSUB 100
<18> 2260 IF fe(x,y)=0 THEN PRINT cue$
<89> 2270 IF fe(x,y)>0 THEN s=s+1:GOSUB 100:PRINT
     inv$;CHR$(149):s=s+3:GOSUB 100:PRINT CHR$(149
     );nrm$:s=s-4
<11> 2280 GOTO 2060
<16> 2290 '
<74> 2300 z=4*y-4:s=8*x-4:GOSUB 100:IF fe(x,y)>0 T
     HEN d=d-fe(x,y)+1:fe(x,y)=0:PRINT SPC(6):z=z+
     1:GOSUB 100:PRINT cue$:z=z+1:GOSUB 100:PRINT
     SPC(6):z=z-1:GOTO 2060
<31> 2310 fe(x,y)=1:PRINT inv$;ste$(1):z=z+1:GOSUB
      100:PRINT ve$;CHR$(149);". ";CHR$(149);ve$:z
     =z+1:GOSUB 100:PRINT ste$(3);nrm$:z=z-1:GOTO
     2060
<96> 2320 '
<21> 2330 IF fe(x,y)=2 THEN fe(x,y)=1:d=d-1:s=s+2:
     GOSUB 100:PRINT inv$;". ";nrm$:s=s-2:GOTO 207
     0
<19> 2340 IF fe(x,y)=0 OR d=2 THEN GOSUB 3020:GOTO
      2070
<49> 2350 fe(x,y)=2:d=d+1:s=s+2:GOSUB 100:PRINT CH
     R$(145);CHR$(148):s=s-2:GOTO 2070
< 9> 2360 '
<19> 2370 '***** Level abspeichern ***************
     *****
<15> 2380 '
< 0> 2390 k=3:k1=3:GOSUB 2720
<39> 2400 IF v(3)=3 OR ta=8 THEN GOSUB 2660:PRINT
     lf1$:GOSUB 3000:GOTO 1750
<47> 2410 IF v(3)=2 THEN 2250
<90> 2420 nstu$="":zus$="":FOR y=1 TO 8:by=0:FOR a
     =0 TO 7:b=2^a:x=a+1
<90> 2430 IF fe(x,y)>0 THEN by=by+b
<40> 2440 IF fe(x,y)=2 THEN zus$=zus$+DEC$(10*x+y,
     "##")
<78> 2450 NEXT a:nstu$=nstu$+RIGHT$("0"+HEX$(by),2
     ):NEXT y
<81> 2460 LSET stu$=nstu$+zus$:IF v(2)=2 THEN PUT
     1,lev ELSE sw=sw+1:PUT 1,sw:lds=sw
<55> 2470 CLOSE 1:GOSUB 2600:GOSUB 2660:PRINT lf1$
     :GOSUB 3000:GOTO 1750
<17> 2480 '
<19> 2490 '***** Unterroutinen ******************
     *****
<94> 2500 '
<60> 2510 z=2:s=9:GOSUB 100:PRINT STRING$(8,138);"
      B O U N C E. -. H I G H S C O R E ";STRING$(
     8,138):z=3:GOSUB 100:PRINT inv$;und$;SPC(51);
     nun$;nrm$
<24> 2520 z=4:GOSUB 100:PRINT lo$;STRING$(22,138)C
     HR$(142)STRING$(7,138)CHR$(142)STRING$(5,138)
     CHR$(142)STRING$(5,138)CHR$(142)STRING$(6,138
     );ro$
<95> 2530 z=5:GOSUB 100:PRINT ve$;" Name des Spiel
     ers:.. ";ve$;" Score ";ve$;" Lev ";ve$;" Bon
     ";ve$;" Sek. ";ve$
<55> 2540 z=6:GOSUB 100:PRINT CHR$(135)STRING$(22,
     138)CHR$(143)STRING$(7,138)CHR$(143)STRING$(5
     ,138)CHR$(143)STRING$(5,138)CHR$(143)STRING$(
     6,138)CHR$(141)
<90> 2550 FOR z=7 TO 27 STEP 2:GOSUB 100:PRINT ve$
     SPC(22)ve$SPC(7)ve$SPC(5)ve$SPC(5)ve$SPC(6)ve
     $:NEXT z
<54> 2560 z=28:GOSUB 100:PRINT lu$STRING$(22,138)C
     HR$(139)STRING$(7,138)CHR$(139)STRING$(5,138)
     CHR$(139)STRING$(5,138)CHR$(139)STRING$(6,138
     )ru$
<80> 2570 FOR i=10 TO 1 STEP -1:z=2*i+6:GOSUB 100:
     PRINT ve$;DEC$(i,"###");". ";na$(i);" ";ve$;D
     EC$(pu(i),"######");" ";ve$;DEC$(le(i),"####"
     );" ";ve$;DEC$(bo(i),"####");" ";ve$;DEC$(pt(
     i),"#####");" ";ve$:NEXT i
```

```
<14> 2580 RETURN
<22> 2590 '
<69> 2600 OPEN "R",1,"BOUNCE.LEV",20:FIELD 1,20 AS
      stu$:RETURN
< 0> 2610 '
< 5> 2620 FOR i=1 TO 10:LSET name$=na$(i):LSET pun
     kte$=DEC$(pu(i),"######"):LSET level$=DEC$(le
     (i),"####"):LSET bonus$=DEC$(bo(i),"####"):LS
     ET dauer$=DEC$(pt(i),"#####"):PUT 2,i:NEXT i:
     CLOSE 2
<56> 2630 OPEN "R",2,"BOUNCE.SCO",35:FIELD 2,16 AS
      name$,6 AS punkte$,4 AS level$,4 AS bonus$,5
      AS dauer$:RETURN
< 9> 2640 '
<91> 2650 z=6:s=82:GOSUB 100:PRINT USING"######";p
     un:RETURN
<66> 2660 z=7:s=79:GOSUB 100:PRINT USING"(###)...
     ";sw:RETURN
<17> 2670 z=7:s=84:GOSUB 100:PRINT USING"####";lev
     :RETURN
<74> 2680 z=8:s=84:GOSUB 100:PRINT USING"####";bon
     :RETURN
<44> 2690 z=9:s=85:GOSUB 100:PRINT USING"###";lim:
     RETURN
<85> 2700 z=10:s=83:GOSUB 100:PRINT USING"#####";p
     ti:RETURN
< 2> 2710 '
<39> 2720 z=19:s=70:GOSUB 100:PRINT lo$;ho$;ro$
<21> 2730 z=20:GOSUB 100:PRINT ve$;inv$;men$(k,1);
     nrm$;ve$
<86> 2740 FOR z=21 TO 19+k1:GOSUB 100:PRINT ve$;me
     n$(k,z-19);ve$:NEXT z
<82> 2750 GOSUB 100:PRINT lu$;ho$;ru$
<69> 2760 v(k)=1:z=20:s=71:GOSUB 3030
<28> 2770 zu$=men$(k,v(k))
<28> 2780 ta=ASC(INPUT$(1)):IF ta<>8 AND ta<>13 AN
     D ta<>21 AND ta<>30 AND ta<>31 THEN 2780
<73> 2790 IF ta=30 THEN v(k)=v(k)+1:IF v(k)=k1+1 T
     HEN v(k)=1
<51> 2800 IF ta=31 THEN v(k)=v(k)-1:IF v(k)=0 THEN
      v(k)=k1
<83> 2810 IF ta=8 THEN IF k>1 AND k<6 THEN PRINT l
     f2$:RETURN ELSE GOSUB 3020:GOTO 2780
<38> 2820 IF ta=13 THEN IF sw>998 AND k=2 AND v(2)
     =1 OR sw<1 AND (k=1 AND v(1)=1 OR k=2 AND v(2
     )=2) THEN GOSUB 3020:GOTO 2780 ELSE PRINT lf2
     $:RETURN
<25> 2830 IF ta=21 THEN GOSUB 2980:GOTO 2780
<20> 2840 GOSUB 100:PRINT zu$:z=v(k)+19:GOSUB 100:
     PRINT inv$;men$(k,v(k));nrm$:GOTO 2770
<16> 2850 '
<88> 2860 bex(1)=VAL(MID$(stu$,17,1)):bey(1)=VAL(M
     ID$(stu$,18,1)):bex(2)=VAL(MID$(stu$,19,1)):b
     ey(2)=VAL(MID$(stu$,20,1)):RETURN
<22> 2870 '
<68> 2880 z=6:s=71:GOSUB 100:PRINT inv$;" Score: "
     ;nrm$:FOR i=1 TO ep1:OUT 248,tw:pun=pun+ep2:O
     UT 248,12:GOSUB 2650:NEXT i:s=71:GOSUB 100:PR
     INT " Score: ":RETURN
<28> 2890 '
<65> 2900 me1=PEEK(&HFBF7):me2=PEEK(&HFBF8):se=VAL
     (HEX$(me1))*60+VAL(HEX$(me2)):IF se>100 THEN
     se=100
< 3> 2910 lim=100-se:GOSUB 2690:pti=ptio+se:GOSUB
     2700
<26> 2920 PRINT lf1$:tz=INT(13*RND)-6:FOR i=1 TO 5
     :FOR j=1 TO 5:z=tz*(i-3)+j+12:s=i*11-1:GOSUB
     100:PRINT pa$(jn(i*5+j-5)):NEXT j,i:GOSUB 303
     0:ta$=INPUT$(1):OUT 245,INT(91*RND):PRINT lf1
     $
< 9> 2930 FOR b=1 TO 8:FOR a=1 TO 8:OUT 246,72-b*8
     -a:IF fe(a,b)>0 THEN OUT 248,tw:s=8*a-4:OUT 2
     48,12:FOR i=1 TO 3:z=4*b-5+i:GOSUB 100:PRINT
     inv$;ste$(i);nrm$:NEXT i
< 3> 2940 NEXT a,b:POKE &HFBF7,me1:POKE &HFBF8,me2
     :OUT 245,91:GOSUB 3030:RETURN
<18> 2950 '
<56> 2960 s=12:FOR z=14 TO 16:GOSUB 100:PRINT MID$
     (bou$,(z-14)*48+1,48):NEXT z:RETURN
<66> 2970 ff=13+3*INT(bw/3):s=(bw MOD 3)*7+70:FOR
     z=ff TO ff+2:GOSUB 100:PRINT MID$(bou$,8*bw+1
     +(z-ff)*48,6);:NEXT z:RETURN
<45> 2980 z1=z:s1=s:SWAP tw,tw1:SWAP so$,so1$:z=11
     :s=85:GOSUB 100:PRINT so$:z=z1:s=s1:GOSUB 100
     :RETURN
<30> 2990 '
<10> 3000 FOR a=1 TO 8:FOR b=1 TO 8:fe(a,b)=0:NEXT
      b,a:RETURN
<38> 3010 s=3:FOR z=1 TO 29 STEP 4:GOSUB 100:PRINT
      eli$:NEXT z:RETURN
<93> 3020 OUT 248,tw:FOR i=1 TO 500:NEXT i:OUT 248
     ,12:RETURN
< 5> 3030 IF INKEY$<>"" THEN 3030 ELSE RETURN
<97> 3040 '
<68> 3050 CLOSE:k=6:k1=4:GOSUB 2720:PRINT clr$;sic
     $;
<77> 3060 IF v(6)=4 THEN OUT 248,1
<20> 3070 IF v(6)=3 THEN PRINT esc$;"x";:SYSTEM
<95> 3080 IF v(6)=2 THEN SYSTEM
<95> 3090 END
```

```
LISTING >LEVEL<, REMARK = >'<.

<36> 100 '********************************************
     ******
<26> 110 '*........ Dieses Programm enthält.....
     .... *
< 3> 120 '*... 100 getestete Level für das Spiel..
     .... *
<45> 130 '*. BOUNCE V 1.4. (C) 12.05.1991 by H. Ge
     yer. *
<57> 140 '*. Jeweils 5 DATA-Werte ergeben einen Le
     vel. *
<46> 150 '********************************************
     ******
< 6> 160 CLEAR ,,,1,20
< 5> 170 OPEN "R",1,"BOUNCE.LEV",20:FIELD 1,20 AS
     stu$
<20> 180 FOR i=1 TO 100:a$="":FOR j=1 TO 5
<66> 190 READ b$:a$=a$+b$:by=by+ABS(VAL("&H"+b$))
< 8> 200 NEXT j:LSET stu$=a$:PUT 1,i:NEXT i
<81> 210 IF by<>7013109! THEN PRINT "Datafehler!"
<28> 220 END
<25> 230 '
<58> 240 DATA 1A54,1A47,0A45,0A50,2127,50D0,CD56,D
     B15
<28> 250 DATA 1000,2557,4000,D457,D114,0F14,3338,4
     2F7
<11> 260 DATA 4A00,42FE,461A,2146,F002,DF12,4362,0
     BF8
<67> 270 DATA 7143,40EE,444C,1F46,A4EC,7367,4D55,4
     805
<66> 280 DATA 2960,A7FA,7334,2227,722A,4A5F,4243,6
     136
<84> 290 DATA 1410,063C,0250,1600,3464,2075,0B09,3
     E2B
<51> 300 DATA 0200,6123,A61C,A7F4,247A,0527,8364,1
     E60
<54> 310 DATA 1CAA,E882,A0AF,2153,80BF,9890,16B8,5
     2D6
< 9> 320 DATA 8157,6171,2766,241A,7050,7112,145F,5
     4C5
< 8> 330 DATA 4FA0,70A0,3477,005A,4A74,742E,5400,7
     336
<78> 340 DATA 08E8,3F40,E95B,086A,2385,0050,4B1D,0
     4C4
<91> 350 DATA 2EF4,2788,BEA8,5DAA,49BA,C995,2143,0
     074
<13> 360 DATA 0E24,2E34,2420,7225,FF89,AD89,8BA5,8
     1FF
<36> 370 DATA 4418,0079,A1C5,A475,042E,1256,283E,0
     8F2
<75> 380 DATA 0812,1E0A,3237,0716,F85A,0602,5EE8,1
     174
<36> 390 DATA 0405,041F,5451,E84D,2447,F507,320E,D
     42C
<98> 400 DATA 80F8,5358,0055,BB51,A20A,7A42,8378,1
     2FB
<40> 410 DATA 12D3,8A18,CF00,1277,A8DD,091C,0911,7
     539
< 3> 420 DATA 7217,0704,1216,3B4A,EB74,1517,AB55,A
     857
<30> 430 DATA 6AF2,4000,2126,84AF,8174,2527,782F,3
     181
<24> 440 DATA C8AA,C80F,C178,E95B,8134,C017,CC99,D
     441
<15> 450 DATA EF41,4427,0055,2A21,FB21,1331,7257,0
     000
<56> 460 DATA B02A,D8D6,E800,2467,0040,EC5A,770A,7
     000
<51> 470 DATA 3346,2A74,6614,2EB4,2E00,5427,0003,0
     057
< 8> 480 DATA 6816,0A05,3418,0503,050A,2CAA,3400,1
     157
<34> 490 DATA 0034,18E4,1C5E,0800,4676,0000,A276,B
     E68
<85> 500 DATA 0028,5425,554E,F52B,11F1,A140,2264,0
     028
```

```
<38> 510 DATA 660A,0C76,182C,2326,7D36,D52A,B902,A
     D06
<72> 520 DATA 6245,302A,142A,060A,7C2A,4228,00F5,C
     B74
< 9> 530 DATA 0B2D,0600,3466,28FF,8061,8348,6A68,2
     264
<37> 540 DATA 0A53,1469,7CC9,34AD,3367,E057,9862,A
     856
<29> 550 DATA C600,3256,5AA7,7194,1BC6,58EC,7334,D
     430
<32> 560 DATA 9620,D635,5263,6427,0517,6518,F4A8,5
     65A
<23> 570 DATA 3375,00DA,5486,6B8D,E00B,3348,006B,1
     659
<20> 580 DATA 420B,5514,3325,63AA,D718,C668,0A00,1
     366
<83> 590 DATA 281D,6358,16CA,A800,2335,0002,5ABA,4
     6B0
<66> 600 DATA 572A,4437,5628,572A,1D62,5D03,3347,0
     050
<36> 610 DATA 10EA,107F,90AC,2476,FF40,A073,2914,0
     D05
<37> 620 DATA 5618,1435,1CE1,1AC9,32A7,6243,4EE2,4
     43A
<76> 630 DATA AA12,A23C,3373,0755,07F8,194B,0557,2
     176
<24> 640 DATA 6058,35CD,A847,5214,1345,520F,5A68,4
     80A
<77> 650 DATA 2A28,5327,A02A,96D0,AF09,340B,7435,4
     3AB
<33> 660 DATA E514,00DA,2AA2,7627,9DE3,8EA2,62BE,6
     A52
<57> 670 DATA 3383,0E0A,635F,2922,5678,1345,0107,1
     B6D
<94> 680 DATA 334A,296D,1446,D50A,8C63,AE51,96AB,1
     145
```

```
<31> 690 DATA 6AC2,B51A,AFD4,37ED,1345,15EB,9825,C
     632
<15> 700 DATA D7AA,1128,1895,C9CE,6BC4,2CA4,2465,5
     7A1
<81> 710 DATA 288F,D801,2DFD,2467,ED49,0A09,DA89,6
     AC3
<64> 720 DATA 2546,E796,A590,9BEC,8780,3136,3EC4,F
     2A9
<95> 730 DATA 137B,6000,4477,605A,36C6,30AF,D085,3
     377
<70> 740 DATA 021B,D7A8,32D6,0A68,3336,00F6,A34A,5
     BA0
<79> 750 DATA DEA8,2255,3A2C,16A8,D65A,6C57,4176,0
     01A
<17> 760 DATA 07B4,2A8B,55D8,1358,46FA,4C12,0897,A
     8B5
<85> 770 DATA 7338,AA1A,90AD,3B86,D080,1586,00A5,D
     639
<35> 780 DATA 271A,3400,4426,2833,DB25,DD71,0414,6
     445
<43> 790 DATA 160C,3A56,6C0E,602C,3568,00A0,B6DA,A
     C66
<18> 800 DATA A8AC,4485,0C28,A6D4,2BF5,0CA8,3448,0
     8ED
<25> 810 DATA 8B48,D4AD,1530,3216,A095,A57A,946E,B
     800
<70> 820 DATA 1267,D428,BC94,16CA,186A,3428,00D5,2
     BEA
< 0> 830 DATA BD4A,7E18,3558,0033,2DDB,35D8,6B0D,1
     386
<93> 840 DATA 0056,2C6A,5218,4AE8,3377,002F,A21D,C
     61B
<87> 850 DATA 5400,2377,1404,1AEC,5A94,56C0,3227,D
     095
<74> 860 DATA 73AC,4A0C,521A,4528
```

Grafiksystem selbst gemacht

Direkte Ansteuerung von Bildschirm, Drucker und Plotter

Beinahe jeder der Leser kam schon mit irgendeinem Grafiksystem in Berührung. Schwierig und mühselig wird es jedoch, wenn das System ein bestimmtes Ausgabegerät, wie zum Beispiel einen Plotter, nicht unterstützt: Es muß ein Treiber geschrieben und installiert werden.
Der Beitrag zeigt, daß es oft weniger aufwendig ist, ein eigenes System zu implementieren. Vor allem, da man dann mehrere Fliegen mit einer Klappe schlagen und eine wichtige und interessante Eigenschaft realisieren kann: eine einheitliche Darstellung auf Monitor, Drucker und Plotter.

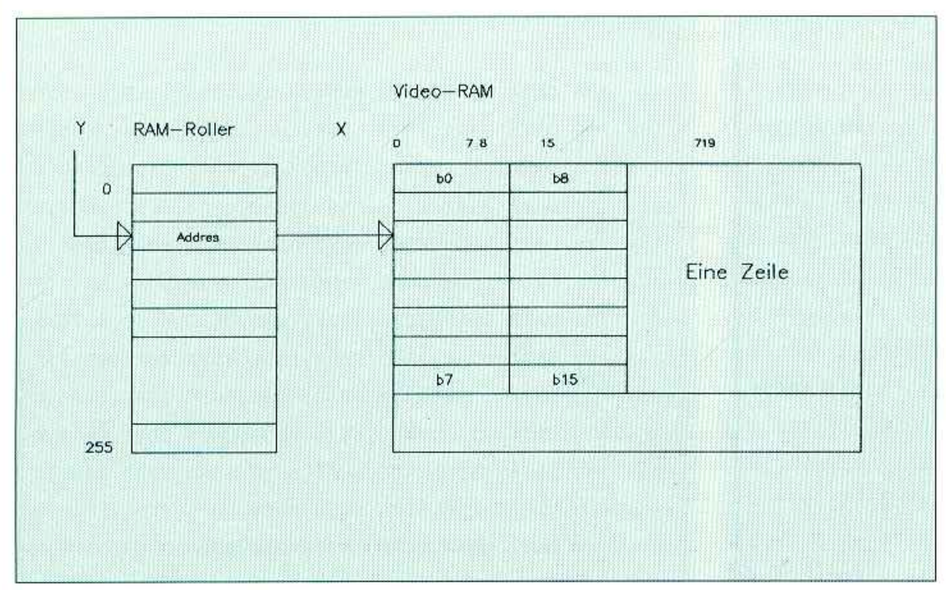
Ein Grafiksystem ist brauchbar, wenn es die dem Anwender notwendigen Funktionen erfüllt und wenn es seine Ausgabegeräte unterstützt. Manche fehlenden Grafikfunktionen können oft mit Hilfe der vorhandenen Funktionen gelöst werden. Schlimmer ist, wenn ein Grafiksystem eines unserer Ausgabegeräte nicht unterstützen kann. Man muß dann einen Treiber schreiben und installieren, und das erweist sich in der Regel, wegen Mangel an vielen notwendigen Informationen, als schwierig.
Deshalb habe ich mich entschlossen, ein einfaches, möglichst übertragbares Grafiksystem zu entwickeln.

Anforderungen an die Grafiksysteme

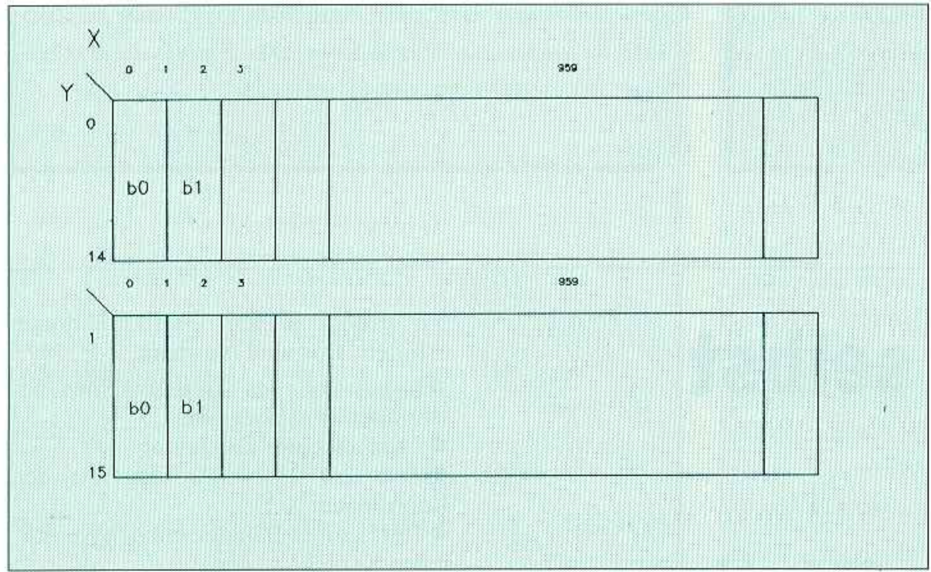
Ein Grafiksystem sollte mindestens, von der Seite des Anwenders gesehen, die grundlegenden Grafikfunktionen erfüllen. Das sind meistens:

- Installation des Treibers eines bestimmten Ausgabegerätes;
- Löschen einer Zeichnung;
- Zeichnen eines Punktes mit den Koordinaten X Y;
- Testen eines Punktes mit den Koordinaten X Y;
- Verschiebung zu einem Punkt mit den Koordinaten X Y;
- Zeichnen einer Strecke oder einer Zackenlinie mit vorgegebenen Parametern;
- Ausfüllen einer Fläche, die durch eine Kontur oder Liste der Koordinaten der Ecken eines Vielecks deklariert ist.

Die oben angegebene Liste kann natürlich noch vervollständigt werden. Man kann noch Farben zugeben, Muster der Linien und Flächen (Ausfüllungen) ändern. Man kann neue Primitiven wie Bögen, Kreise, Rechtecke oder Texte zugeben. Man kann nicht nur zeichnen, sondern auch löschen oder bereits gezeichnete Punkte modifizieren. Das alles war aber nicht unser Ziel. Beachten wir doch, daß, obwohl das Hinzufügen von diesen Funktionen einfach ist (wir denken hier, daß jeder Leser dazu fähig wäre), sofort gewisse Probleme auftauchen. Nicht alle Ausgabegeräte sind fähig, zum Beispiel die Zeichnungen in



Die Speicherverteilung des Monitortreibers



Nach diesem Prinzip wird die Speicherverteilung des Druckertreibers durchgeführt.

Farbe zu bilden. Unmöglich ist auch zum Beispiel das Löschen einer einmal gezeichneten Linie mit Hilfe des Plotters. Bei einer genaueren Analyse kann man noch mehr ähnliche "Probleme" entdecken. Zusätzliche Schwierigkeiten bereiten verschiedene Auflösungen und Formate der Zeichnungen und (was besonders kläglich ist) verschiedene Pixelproportionen. Die Voraussetzung war dagegen das Erhalten von möglichst identischen Zeichnungen bei Anwendung der Ausgabegeräte verschiedener Typen. Unter dieser Voraussetzung wurde ein Programmpaket entwickelt (Listing 1.). Es realisiert die vorher aufgeführten Funktionen mit Hilfe der Unterprogramme

```
OpenDevice, CloseDevice, Point, Dot,
Plot, Draw, Line, Fill, Area.
```

Man sollte auch Reaktionen des Grafiksystems im Falle der Überschreitung des effektiven Bereiches der Zeichnung bestimmen.

Wir haben in diesem Falle Arbeit in den Regimen Abschneiden, Rollen oder Unterbrechung des Zeichnens (ähnlich wie in der Programmiersprache Logo) zugelassen. Korrekt konstruierte Prozeduren zum Abschneiden und Rollen sollten auch einwandfrei im Falle eines verkleinerten Formats des Bildes (Fenster) arbeiten.

Ausgabegeräte und ihre Treiber

Typische Ausgabegeräte für ein Grafiksystem sind die Bildschirme, Drucker und etwas seltener Plotter. Auf diese Geräte wurde beim Schreiben von Beispieltreibern besonders besonders geachtet. Es wurde angenommen, daß jedes der Ausgabegeräte ziemlich "unintelligent" ist. Das bedeutet, daß im Falle eines Zugriffs auf den Bildschirm der Zugriff nur auf den Bildspeicher gestattet ist, im Falle des Druckers die Möglichkeit der direkten Steuerung der Druckernadeln verfügbar ist, und bei dem Plotter das Zeichnen einer Linie zwischen zwei Nachbarpunkten und Verschiebung des Plotterköpfes in gezeigter Stellung möglich ist. Selbstverständlich bieten meistens die grafischen Geräte mehr Möglichkeiten an. Bei dem Monitor besteht oft die Möglichkeit der Ausnutzung von Systemprozeduren, die die ausgewählten Grafikfunktionen realisieren. Matrixdrucker bieten häufig eine ganze Menge von Grafikfunktionen, manchmal besitzen sie auch Plotteremulationen (was möglich ist, falls der Drucker die Welle in beiden Richtungen drehen kann). Plotter sind in der Regel ziemlich intelligente Geräte. Sie ermöglichen nämlich außer dem Zeichnen von Linien, auch die Ausgabe von Texten und manchmal auch eine Ausfüllung von Vielecken. Die Ausnutzung solcher zusätzlicher Möglichkeiten vereinfacht den Aufbau einiger Fragmente des Treibers.

Unser Treiber soll folgende Funktionen erfüllen:

- Zeichnen des Punktes mit den Koordinaten X Y;
- Testen des Punktes mit den Koordinaten X Y;
- Verschiebung zum Punkt mit den Koordinaten X Y;
- Löschen einer Zeichnung.

Die übrigen grafischen Funktionen, solche wie Zeichnen einer Linie oder Ausfüllen einer Fläche, sollten schon durch entsprechende Prozeduren des Grafiksystems realisiert werden.

Jetzt wären die Möglichkeiten der einzelnen Grafikgeräte an der Reihe. Der Monitor des Amstrad PCW besitzt die Auflösung 720 x 256. Der Bildspeicher befindet sich in der Speicherbank 0, zu dem man mit Hilfe der XBIOS-Funktion Nummer $E9 Zugriff erhalten kann. Die Adresse des Bytes, das den vorgegebenen Punkt enthält, wird anhand der X,Y-Koordinaten und des Inhaltes des sogenannten RAM-Roller-Bereiches, das die Anfangsadressen jeder der 256 Bildschirmlinien [1] sammelt, berechnet. Diese Berechnung wird durch die Funktion *Adr* ausgeführt. Die einzelnen Bytes im Bildschirmspeicher sind so organisiert, daß Entsendung zu ihnen der aufeinanderfolgenden Acht-Bytes-Zeichenmatrizen eine Textzeile auf dem Bildschirm bildet.

Der Computerdrucker ist (teilweise) mit dem Epson FX kompatibel. Zur Bildung der Zeichnung wurde eine höhere Bildqualität gewählt, die der Auflösung von 960 Punkten in einer Linie entspricht. Ich nahm 720 Linien an, damit ein für den Bildschirm typisches Verhältnis der Breite zu der Höhe des Bildes von vier zu drei gewährleistet wird. Diese Auflösung bestimmt die Größe des Bildspeichers von 90 kByte. Das Vorhandensein einer ziemlich großen RAM-Disk im Computer führte zu der Lösung, in welcher eine ensprechend große Datei mit direktem Zugriff den Bildspeicher für den Drucker emuliert. Es sollte eine einschlägige Struktur dieser Datei angenommen werden. Die Datei wurde in Blocks zu je einem kByte geteilt. Jeder Block nutzt jedoch nur 960 Bytes aus, was acht Linien des Bildes entspricht. Zur Beschleunigung der Ausgabe der Zeichnung auf den Drucker wurde angenommen, daß die aufeinanderfolgenden Bytes in der Datei die Information bilden, die direkt an den Druckkopf des Druckers ausgegeben wird. Es bedeutet, daß alle Bits jedes Bytes der selben X-Koordinate entsprechen. Um eine höhere Auflösung zu erreichen, entsteht jede Zeile der Zeichnung (16 Linien) durch zweimaliges Passieren des Druckkopfes. Aus diesem Grunde enthalten die Blocks mit geraden Nummern die Information über Punkte mit ungeraden Y-Koordinaten und umgekehrt. Eine solche Lösung beschleunigt bedeutend die Ausgabe der Zeichnung.

Die Umrechnung der X,Y-Koordinaten auf die Blocknummer, Adresse im Block und Bitnummer erfolgt in der Prozedur *Calculate*. Eine weitere interessante Lösung ist der Datenaustausch mit der Datei mit Bitkarte. Der klassische direkte Zugriff zur Datei kann hier nicht angenommen werden, weil das Zeichnen von senkrechten Linien laufende Ein- und Ausgaben des Dateipuffers verursacht hätte. Man entschied sich also für das Modell des virtuellen Speichers aus der Programmiersprache Forth.

In jedem Augenblick befinden sich im Arbeitsspeicher immer zwei Blocks. Eine Forderung des Zugriffs auf einen bestimmten Block verursacht die Überprüfung, ob sich der Block mit der angegebenen Nummer im Arbeitsspeicher befindet. Falls ja, erfolgt kein Einlesen des Blocks von der Disk. Sonst wird der geforderte Block von der Disk in den Puffer eingelesen. Um das jedoch durchführen zu können, muß man vorher den Puffer durch Löschen aus dem Speicher des am längsten verwendeten Blocks freimachen. Wenn dieser Block vorher modifiziert wurde, wird er auf die Disk geschrieben. Den virtuellen Zugriff zum Speicher realisiert die Funktion *Block*.

Das letzte angewendete Gerät war der Plotter Sharp CE. Er kann Linien zeichnen und Texte ausgeben. Der von mir erstellte Treiber nutzt jedoch nur die Möglichkeit des Vorschubs des Plotterkopfes und Verbindung von zwei benachbarten Punkten mit einer Linie aus. Das Problem bildet beim Plotter die Ausfüllung, weil der Plotter den Punkt mit vorgegebenen Koordinaten nicht testen kann. Die Lösung dieses Problems erforderte die Reservierung einer ensprechenden Datei mit Bitkarte auch für Plotter. Weil die Auflösung des Plotters in der horizontalen Achse 960 Punkte betrug, war die Annahme der gleichen Auflösung wie bei dem Drucker möglich. Ähnlich ist auch die Größe der Datei mit der Bitkarte, die 90 kByte beträgt. Jeder Block sammelt nämlich Informationen über den Inhalt von acht aufeinanderfolgenden Bildlinien. Die folgenden Bits jedes Bytes entsprechen den nächsten Werten der X-Koordinate. Es vereinfacht und beschleunigt die Ausfüllung, welche der Plotter durch Straffierung mit horizontalen Linien realisiert. Die Umrechnung der X,Y-Koordinaten auf die Blocknummer, Adresse im Block und Bitnummer erfolgt in der Prozedur *Calculate*. Auch im Falle des Plotters wurde der vorher beschriebene Mechanismus des virtuellen Zugriffs auf die Datei mit der Bitkarte angewendet, obwohl in diesem Falle der Nutzen nicht so augenscheinlich wie beim Drucker ist.

Alle drei Treiber wurden als Überlappungsstrukturen (Overlays) implementiert, weil in jedem Augenblick nur einer von ihnen ausgenutzt wird. Diese Treiber sind in den Listings 2, 3 und 4 dargestellt.

Unter Ausnutzung der durch die Treiber angebotenen Möglichkeiten kann man weitere Funktionen des Grafiksystems entwickeln.

Die erste von ihnen ist die Prozedur *Draw*, die eine Strecke zeichnet. Sie nutzt den vielmals beschriebenen Algorithmus von Bresenham, deshalb wird Sie hier auch nicht genauer angesprochen.

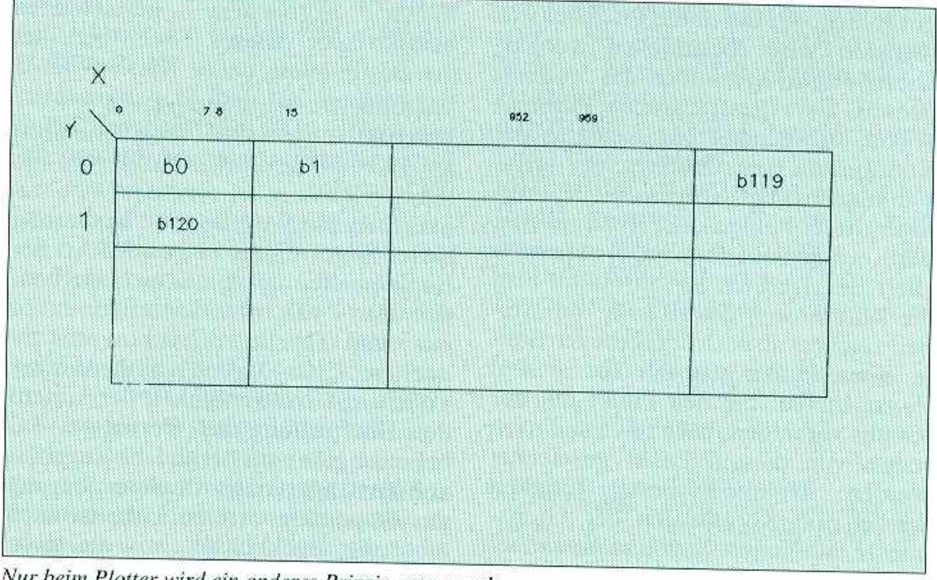
Bemerken Sie bitte nur, daß diese Prozedur sich leicht so modifizieren läßt, daß statt einer stetigen eine Linie nach vorgegebenem Muster gezeichnet wird. Leider werden die horizontalen und senkrechten Linien wegen der Pixelproportionen unterschiedlich sein (besonders auf dem Bildschirm). Eine Vervollständigung der Prozedur *Draw* bildet die Prozedur *Line*, die eine Zackenlinie ausgibt.

Angewendete Algorithmen

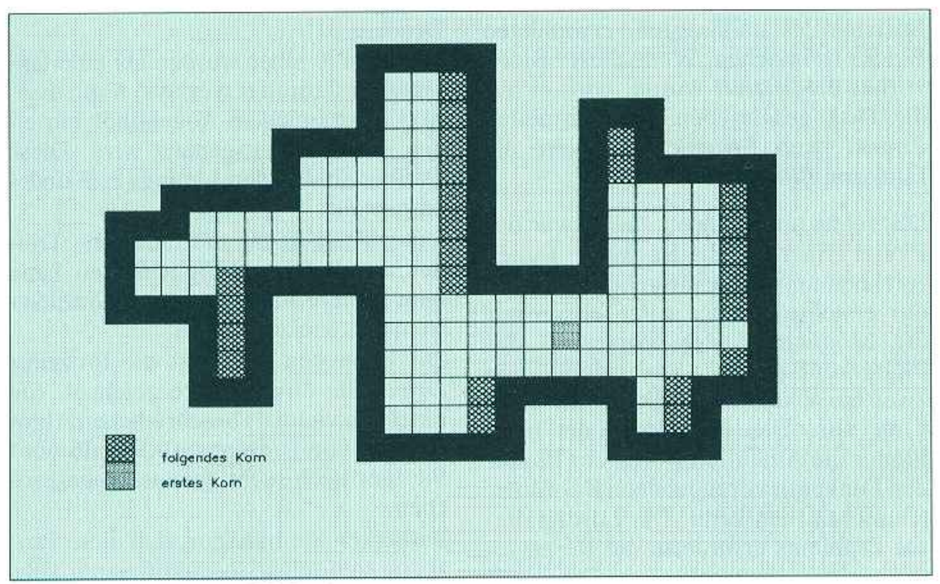
Viel interessanter ist das Problem der Ausfüllung von Flächen. Die Aufgabe der Ausfüllung von Flächen kann wie folgt definiert werden:

- Bemalung einer Fläche, deren Kontur (nicht unbedingt regulär) vorgegeben ist.
- Bemalung des Inneren eines Vielecks (nicht unbedingt konvex), dessen Eckpunktkoordinaten gegeben sind.

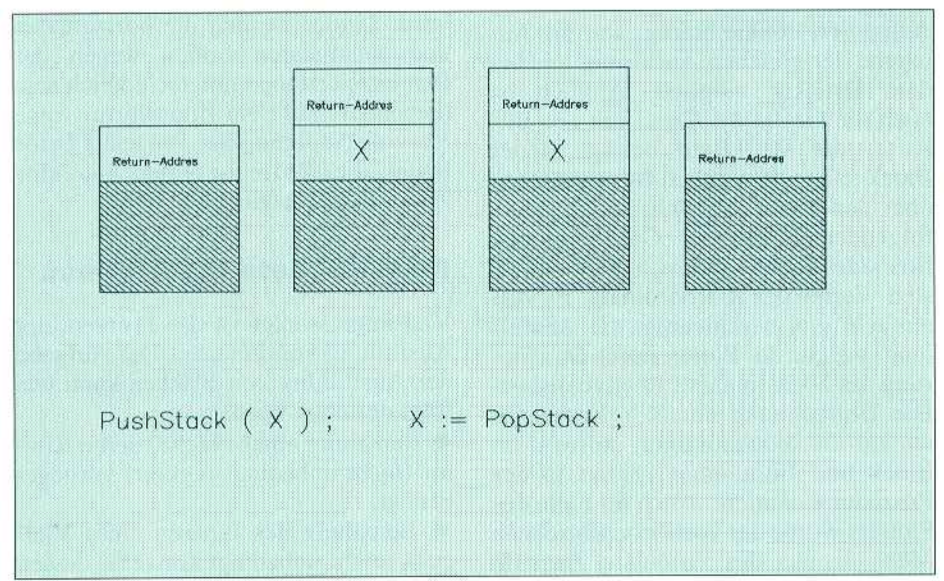
Das vorgestellte Paket enthält die Lösung dieser beiden Aspekte auf eine,



Nur beim Plotter wird ein anderes Prinzip angewandt



Die Arbeitsweise unserer Fill-Routine



Das Parameter-Spiel bei PushStack und PopStack

wie wir meinen, interessante Weise. Zuerst sollte jedoch der erste dieser Algorithmen, der durch die Prozedur *Fill* realisiert wird, besprochen werden. Dieser Algorithmus setzt das Vorhandensein einer geschlossenen Flächenkontur im Speicher und Kenntnis der Koordinaten eines Punktes, der in ihrem Inneren liegt, voraus. Die Koordinaten dieses Punktes, der Korn genannt wird, werden auf dem Stapel abgespeichert. Im nächsten Schritt nimmt man die Koordinaten des Kornes vom Stapel, zeichnet an der vorgegebenen Stelle einen Punkt, wonach aus diesem Punkt Linien, die den Punkt mit der Kontur verbinden, nach links und nach rechts von diesem Punkt gezeichnet werden. Während dieser Tätigkeit werden die Koordinaten der Grenzpunkte der Kontur *Xleft* und *Xright* bestimmt. Dann wird die Prozedur *Scan* aufgerufen, die die Linie testet, die über und unter der durch *Xleft* und *Xright* begrenzten, gezeichneten Strecke liegt. Dieses Testen bedeutet die Bestimmung neuer Körner, deren Koordinaten auf dem Stapel geschrieben werden. Die beschriebenen Tätigkeiten werden wiederholt, bis der Stapel leer wird.

Die Schwierigkeit bei der Kodierung dieses Algorithmus ist mit der Art der Implementierung des Stapels verbunden. Am einfachsten kann man ihn in einer Art Tabelle realisieren, was jedoch seine Größe von vornherein begrenzt, die Notwendigkeit der laufenden Überprüfung des Wertes des Tabellenzeigers einführt und die Zugriffszeit verlängert. Die Implementierung des Stapels durch eine Liste ist auch nicht die glücklichste, weil sie mehr Speicherplatz erfordert (jedes Listenelement enthält zusätzlich einen Zeiger) und die Zugriffszeit verlängert. Ideal wäre wegen der Speicherbelegung und Zugriffszeit die Ausnutzung des Maschinen-Stacks (des Prozessors Z80). Das Problem besteht jedoch darin, daß der Prozessor Z80 nur einen Stack (Stapel) besitzt, dessen Zeiger sich im SP-Register (Stack Pointer) befindet, und daß dieser Stapel von Turbo-Pascal für die Aufbewahrung der Rückkehradressen und der Parameterübergabe ausgenutzt wird. Auch dieses Problem konnte ich jedoch mit Hilfe der Prozedur *PushStack* und der Funktion *PopStack* lösen. Die Prozedur PushStack lädt seinen Parameter auf dem Stapel unter ihrer Rückkehradresse, seinen Wert entnimmt die Funktion *PopStack* dagegen vom Stapel aus der Stelle unter der Rückkehradresse. Bei Benutzung der Prozedur *PushStack* und der Funktion *PopStack* sollte man etwas vorsichtig sein. In der Regel sollten alle Operationen auf dem Stapel innerhalb einer Prozedur durchgeführt werden. Wenn wir diese Operationen in einer anderen Prozedur ausnutzen möchten (so wie in der Prozedur Scan), so muß diese Prozedur mit der Anweisung:

```
Return := PopStack ;
```

beginnen und mit der Anweisung:

```
PushStack (Return) ;
```

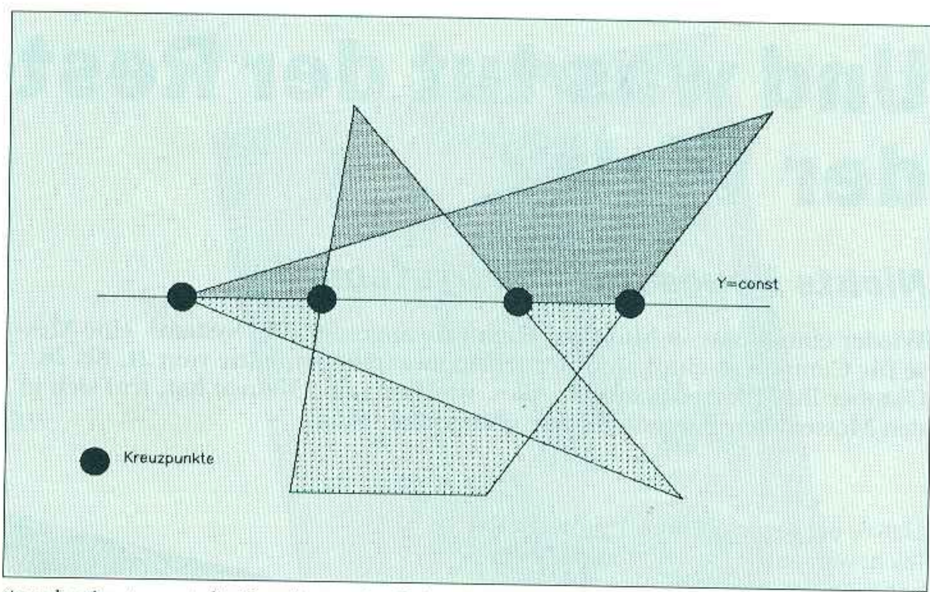
enden, was ein Vermischen der Adressen und Daten auf dem Stapel verhindert. Man sollte auch in solchen Prozeduren die Anweisung *For* , die die Operationen *Push-Stack* und *PopStack* benutzt, nicht anwenden, weil diese Anweisung den Maschinenstapel benutzt.

Im Falle anderer Prozessoren mit einer größerer Anzahl von Adreßregistern, aus denen jedes als Stapelzeiger angewendet werden kann, vereinfacht sich das beschriebene Problem wesentlich.

Der zweite der Ausfüllalgorithmen, der durch die Prozedur *Area* realisiert wird, erfordert die Bestimmung der Koordinaten der Eckpunkte des zu bemalenden Vielecks. Er erfordert kein Vorhandensein einer Bildbitkarte, deshalb wird die Ausfüllung diesen Typs durch eingebaute Kommandos bei einigen Plottern realisiert. Dieser Algorithmus überprüft zuerst, ob die Kontur geschlossen ist (ob Koordinaten des letzten Eckpunkts identisch mit den des ersten sind). Falls nicht, so wird die Kontur durch Zufügen eines nächsten Eckpunktes geschlossen. Dann wird die Kontur mit Hilfe der Prozedur *Line* gezeichnet. Im nächsten Schritt bestimmt man den minimalen und maxi-

malen Wert der Y-Koordinate durch Durchsuchen der Eckpunktliste. Dies führt die Prozedur *RangeY* aus. Jetzt bestimmt man für jede Y-Linie, die zum gegebenen Intervall gehört, die X-Koordinaten der Schnittpunkte mit allen Strecken, die zur Kontur gehören. Diese Koordinaten werden durch die Funktion *Xcross* berechnet. Man kann beweisen, daß, falls die Kontur geschlossen ist, die Anzahl ihrer Schnittpunkte mit der Gerade Y=const gerade ist. Um eine solche Kontur zu bemalen, muß man also die zuvor bestimmten Punkte entsprechend verbinden. Das Problem im diesem Falle liegt darin, daß die X-Koordinaten der Schnittpunkte nicht der Reihe nach bestimmt werden. Deshalb müssen die erhaltenen Koordinaten vor Beginn der Operation der Punkteverbindung sortiert werden. Weil wir wiederum von vornherein nicht wissen, wieviele Schnittpunkte wir erhalten, werden ihre X-Koordinaten üblich in einer Listenstruktur abgespeichert. Um diesen Algorithmus zu verbessern, werden nacheinander bestimmte X-Koordinaten im Maschinenstapel abgespeichert, wobei der Stapel jedesmal bei Eintragung einer neuen Zahl sortiert wird. Die Sortierung des Stapels wird in der Prozedur *SortStack* durchgeführt. Um diesen Prozeß zu optimieren, wurde ein zweiter Stapel angewendet. Einen Zugriff auf ihn gestatten die Prozedur *PushHeap* und die Funktion *PopHeap*. Dieser Stapel wächst in entgegengesetzter Richtung zum Vergleich mit dem Maschinenstapel. Die Sortierung wird selbst durch Einschiebung realisiert. Sie beruht auf Wegnehmen vom Stapel aller Koordinaten, die kleiner als die eingeschobene Koordinate sind. Diese vom Maschinenstapel weggenommenen Koordinaten werden zeitweise auf dem zweiten Stapel abgelagert. Dann wird die eingeschobene Koordinate auf den Maschinenstapel geschrieben und alle vorher auf dem zweiten Stapel abgelagerten Koordinaten werden zurückgeschrieben. Diese Lösung ermöglicht die Ausnutzung des gesamten freien Hauptspeichers zum Sammeln der X-Koordinaten der Schnittpunkte. Nach der Bestimmung und Sortierung aller Schnittpunkte verbindet die Prozedur *DrawScan* durch Strecken die Punkte, deren X-Koordinaten auf dem Stapel abgespeichert sind, wodurch das Vieleck bemalt wird.

Üblicherweise erscheint am Ende eines solchen Artikels eine Beispielprozedur oder ein Demonstrationsprogramm.



Area bestimmt zuerst die Koordinaten der Eckpunkte des auszumalenden Rechtecks

Beispiel der Anwendung

Endgültig wählte ich eine Prozedur, die (pseudo-) dreidimensionale Zeichnungen der Funktionen von zwei Variablen bildet. Diese Prozedur, genannt *FnGraph*, sucht selbst die Art des installierten Ausgabegerätes aus, paßt ihm die Auflösung der Zeichnung an und skaliert die Zeichnung. Die Parameter der Prozedur FnGraph bildet die Funktion, deren Zeichnung wir ausgeben wollen, und der Bereich, in dem die Zeichnung abgebildet werden soll. Dieser Bereich ist ein Würfel, der durch die Werte der Parameter Xmin, Xmax, Ymin, Ymax, Zmin und Zmax begrenzt ist. Einige Schwierigkeiten macht im Turbo-Pascal die Übergabe der Funktion als Parameter (Turbo-Pascal läßt keine Parameter zu, die Prozeduren sind). Dieses Problem wurde bereits in [3] [4] gelöst. Als Parameter wird zur Prozedur der Wert der Adresse der Funktion, die den Parameter bildet.

Diese Adresse kann man mit Hilfe der Funktion *Addr* erreichen. Innerhalb der Prozedur FnGraph wird der Aufruf der den Parameter darstellenden Funktion durch die Funktion *CallFunc* realisiert. Dies ist eine Ein-Parameter-Funktion, die durch Inline-Anweisung realisiert wurde. Ihr Parameter bestimmt die Adresse, die man anspringen soll. Es bleibt noch das Problem der Parameterübergabe. Turbo-Pascal übergibt die Parameter auf dem Stapel, wobei durch ihn jedesmal nach Eingang in die Prozedur oder Funktion automatisch eine Anweisungssequenz generiert wird, die die Parameter vom Stapel löscht. Ausnahme bilden alle als externe deklarierte (external) Prozeduren, die das selbst machen sollten. Diese Tatsache nutzte ich bei Deklaration der externen Prozedur *PushReal* aus. Diese ist in Wirklichkeit keine externe Prozedur, und ihr Code bildet nur die Maschineninstruktion RET (Rückkehr aus dem Unterprogramm). Ein Aufruf dieser Prozedur verursacht also im Endeffekt nur eines, daß nämlich der Parameter auf dem Stapel gelassen wird. Diese Parameter werden dann durch die Funktion ausgenutzt, die wiederum Parameter der Prozedur *FnGraph* ist. Ein weiteres Problem, das man bei der Konstruktion solcher Zeichnungen antrifft, ist das Verdecken der durch den Beobachter nicht gesehenen Zeichnungsfragmente. Die Prozedur FnGraph löst es ganz einfach durch das Vergleichen von zwei benachbarten Zeichnungskurven.

Literatur:

[1] Andrew R. M. Clarke, David Powys-Lybbe, The Amstrad CP/M Plus, M.M.L. Systems Ltd., London 1986.

[2] Michael Anton, Im Herzen von CP/M Plus - XBIOS zerlegt, Joyce-Sonderheft 1'87.

[3] Namir Clement Shammas, Turbo-Pascal Procedural Parameters, Software Tools December 86.

[4] Bernd Rüffer, Noch ein Turbo-Tip - Prozedurale Parameter in Turbo-Pascal, C'T September 86.

(Zbigniew Szkaradnik/rs)

Aus Platzgründen haben wir darauf verzichtet, die Listings zu diesem Artikel abzudrucken. Auf der PCW Databox finden Sie jedoch eine Umsetzung der im Artikel beschriebenen Software.

Und was tut der Rest der Welt?

Nichts Neues auf der Systems

Wieder einmal war in München High Life angesagt: Die Systems, eine Messe für Computertechnik und Branchenanwendungen, hatte vom 21. bis 26. Oktober ihre Pforten geöffnet. Alles, was Rang und Namen hat, traf sich in den Messehallen Bayerns liebster und größter Stadt.

Daß es bei so viel Gerangel gar nicht so leicht war, auch etwas für unsere "Kleinen" an Informationen zu ergattern, versteht sich von selbst. Zwar fand man hier und da bei genauerem Hinsehen noch so manches an Interessantem, doch es bleibt Fakt: Auf der ganzen Systems wurde kein einziger CPC ausgestellt. Lediglich auf dem AMSTRAD-Stand konnte man bei der Firma Wiedmann, die sich als Nebenaussteller präsentierte, einen PCW vorfinden.

Daß dieser jedoch bis über beide Ohren – oder sagen wir besser bis über den Tastaturanschluß – vollbeladen mit Zusätzen war, kann sicherlich über diese Tatsache hinweghelfen.

Ansonsten gestaltete sich die Systems recht ruhig. Man merkte aber, daß der Trend in Sachen Speichermedien immer weiter in Richtung der optischen Speicherplatten geht. Dies sollte jedoch aufgrund der horrenden Preise für die CPC- und PCW-Benutzer nur nebenbei erwähnt sein.

Ralf Schößler





Geschäftiges Tun am Amstrad-Stand. Herr Wiedmann war immer bemüht, den Besuchern mit Rat und Tat zur Seite zu stehen. Ob es nun um den PCW, den AMSTRAD PC mit drei Laufwerken oder aber um LocoScript PC ging – bei der Firma Wiedmann war man richtig.

Gerade jedoch für die PCW-Besitzer war es sicher mehr als interessant, einmal den PCW zu beschnüffeln. Gerade jetzt, wo die neue Version von LocoScript auf den Markt gekommen ist, konnte man sich so ein Bild von der ganzen Sache machen.

Leider war es Herrn Wiedmann nicht möglich, einen der neuen PCWs, die auf der Business Computing '91 in England erstmalig der Öffentlichkeit vorgestellt wurden, auf der Münchner Messe auszustellen.

Am Informationspult des AMSTRAD-Standes lagen alle Informationen zur neuen und zu schon bewährter AMSTRAD-Technik aus. Neu zur Systems war zum Beispiel "das gemischte Doppel" des AMSTRAD-Laptop ALT 386 SX in Verbindung mit dem Canon BubbleJet 10e.

In einem geordneten … findet sich alles, oder so ähnlich könnte man den Spruch abwandeln. Erst nach längerem Suchen war es den Herren vom Interest-Verlag möglich, uns ein Exemplar des Sammelbandes "Mehr Erfolg mit CPC" zu präsentieren.



Trotzdem, oder gerade deshalb geschah dies dann jedoch mit der doppelten Begeisterung, was bei einem Mammut-Werk wie diesem sicherlich nicht verwundern dürfte. So findet man innerhalb dieses Sammelordners alles über die Hard- und Softwareseiten des CPC.

AMSTRAD wartete diesmal zur Systems wieder mit einem neuen Stand auf. Auch an der Größe wurde sichtlich gefeilt. Man könnte meinen, daß die 4er Serie des AMSTRAD einen wahren Geldsegen beschert habe. Verdient wäre es auf jeden Fall.



Minis in Reih und Glied konnte man am AMSTRAD-Stand bewundern. Der Trend geht immer mehr zu Computern, die durch ihren geringen Platzbedarf auf dem Schreibtisch auffallen. Mit dem AMSTRAD Mini PC 4386SX sowie dem 5286 dürfte es gelungen sein, hier das minimal Mögliche in die Tat umzusetzen.

Für Sie gespielt

Trivial Pursuit und Scrabble auf dem PCW

Wer immer noch der Meinung ist, daß es für den PCW keine Spielesoftware gibt, sollte sich ganz schnell eines Besseren belehren lassen. Wir stellen Ihnen wieder zwei neue Programme vor, die es sowohl für den CPC als auch den PCW gibt.

Bei beiden Programmen handelt es sich um Umsetzungen von ursprünglich als Brettspiel entwickelten Spielen. Da auch hier mit mehreren Teilnehmern gespielt werden kann, stellen diese Programme eine Bereicherung für jede Spielesammlung dar. Spaß für die ganze Familie.

Trivial Pursuit

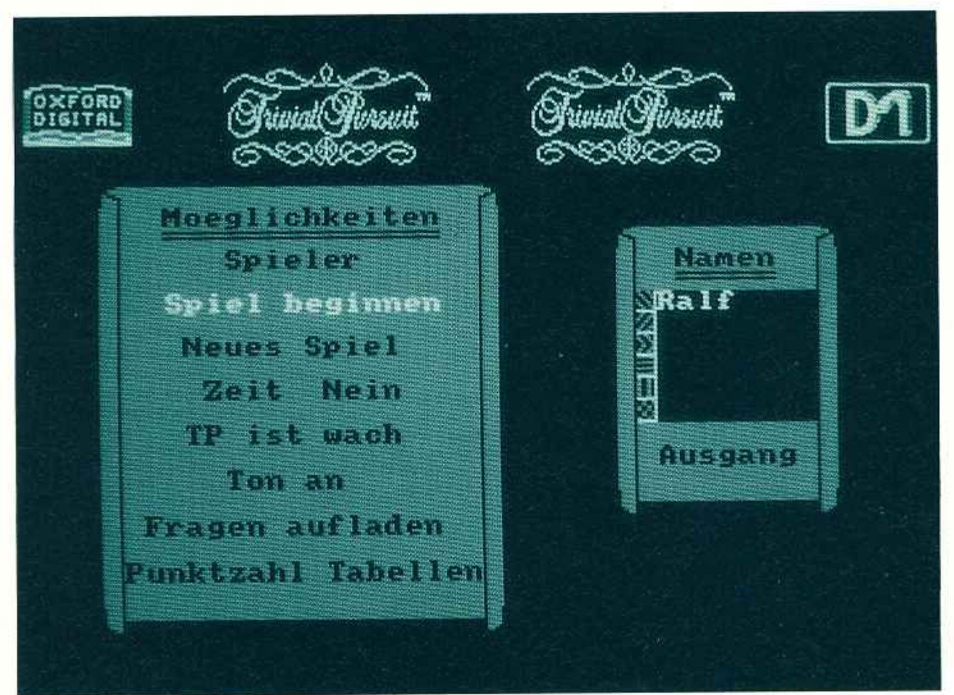
Grundsätzlich geht es bei Trivial Pursuit darum, die vom Computer gestellten Fragen aus den Bereichen Geschichte, Wissenschaft und Natur, Erdkunde, Unterhaltung, Sport und Literatur richtig zu beantworten. Auf sechs Feldern des Bretts besteht die Möglichkeit, sich sogenannte Wissensecken der sechs Themengebiete zu erwerben. Nachdem man alle Wissensecken gesammelt hat, muß man zum Mittelpunkt des Bretts ziehen, um dort die Schlußfrage richtig zu beantworten. Die Fragen der einzelnen Themengebiete werden von einem kleinen Männchen, das als Quizmaster fungiert, gestellt, wobei auch Musikfragen im Angebot sind: Der JOYCE fängt tatsächlich mit seinen begrenzten Möglichkeiten an, ein Musikstück vorzuspielen, zu dem es dann die gestellte Frage zu beantworten gilt. Wie auch beim Brettspiel, muß die Antwort nicht aufgeschrieben oder eingetippt werden.

Schummeln gilt nicht!

Die Mitspieler entscheiden, ob die gegebene Antwort in etwa der gesuchten entspricht, die vom Quizmaster nach einer einstellbaren Zeit (fünf Sekunden bis neun Minuten oder unbegrenzt) preisgegeben wird. Daß das komplett in deutsch gehaltene Spiel von seinen Programmierern von Oxford Digital Enterprises gut durchdacht wurde, zeigt sich auch in der Spielerverwaltung. Zunächst einmal können bis zu sechs Personen am Spiel teilnehmen. Man kann das Spiel natürlich auch alleine spielen, wobei meiner Meinung nach jedoch der gesellschaftliche Reiz, den dieses Spiel nun einmal ausübt, verlorengeht. Wenn man das Spiel mit mehreren Personen spielt, ist es möglich, daß ein Spieler ausscheiden oder Pause machen kann (besonders wichtig für Spieler mit schwachen Nerven, die sich erst einmal von der Aufregung des Spiels erholen müssen). Anschließend kann er selbstverständlich wieder in das Geschehen einsteigen, was auch für Nachzügler gilt, die, begeistert vom Spiel, neu beginnen wollen.

Um seinem Mitspieler zu zeigen, daß man das Wissensgenie überhaupt ist, besteht die Möglichkeit, jederzeit eine Punktetabelle aufzurufen, in der richtige und falsche Antworten statistisch aufgeführt sind. Wenn man alle im Speicher befindlichen Fragen schon einmal beantwortet hat, kann man selbstverständlich neue Fragen nachladen, so daß lange Weile durch immer gleiche Fragen bestimmt nicht aufkommt. Tastatur- oder Joystick-Freaks kommen gleichermaßen auf ihre Kosten, da das Programm über die Tastatur einstellbar oder über das Joystick-Interface von Cascade/Kempston steuerbar ist.

Da die Bildschirmdarstellung mit hervorragender Grafik gestaltet wurde, fällt auch die letzte Möglichkeit eines eventuellen Schwachpunkts weg, so daß letztendlich nur die eine Wertung in Frage kommt: Das Spiel ist spitze. Der Preis von DM 89,50 hat seine volle Berechtigung, die Programmierer von Oxford Digital Enterprises haben für dieses Geld 100prozentige Arbeit geleistet.



Das Trivial-Pursuit-Menü: Hier können auch während des Spiels Einstellungen getätigt werden

Scrabble

Wie auch beim original Scrabble, erhält man vom Computer sieben Buchstaben, denen Punkte von eins bis zehn zugeordnet sind.

Nun gilt es, ein Wort aus diesen Buchstaben zu bilden oder auf dem Brett befindliche Wörter sinnvoll durch hinzufügen neuer Buchstaben zu ergänzen, jeweils mit dem Ziel, eine möglichst hohe Punktzahl zu erreichen. Durch Felder wie "Dreifaches" oder "Doppeltes Wort" und "Dreifacher" oder "Doppelter Buchstabenwert" können die Punkte für das neu gebildete Wort noch erhöht werden. Wenn man dann von den sieben Buchstaben zum Beispiel fünf gelegt hat, bekommt man vom Computer fünf zufällig ausgewählte neue Buchstaben hinzu, so daß man für die nächste Runde wieder sieben hat.

Das Spiel ist beendet, wenn alle Spieler keine neuen Wörter mehr bilden kön-

nen. Dies ist zumeist dann der Fall, wenn der Vorrat des Computers von 100 Buchstaben aufgebraucht ist. Gewonnen hat, wer die meisten Punkte gelegt hat.

20 000 Wörter

Die Umsetzung dieses beliebten Brettspiels auf den Computer könnte man im Prinzip als durchaus gelungen bezeichnen, wenn nicht die für den deutschen Markt wohl wesentliche Einschränkung existieren würde: Das Spiel ist komplett in Englisch gehalten und auch die Tastatur wurde auf das englische Layout umgeschaltet. Daß die Bildschirmausgaben englisch sind, läßt sich durchaus noch verschmerzen. Schwierig wird es jedoch spätestens in dem Moment, wenn man gegen den Computer antritt. Dieser hat nämlich einen umfangreichen Wortschatz von zirka 20 000 ausgewählten Wörtern, die aber eben alle aus dem englischen Sprachgebrauch stammen. Der deutsche Spieler, der gegen den Computer antritt, hat es vermutlich sehr schwer, da sein englischer Wortschatz, wenn überhaupt, nur zirka 1500 bis 2000 Wörter umfaßt. Dies ist definitiv zu wenig, um damit gewinnen zu können. Die Hilfsfunktion des Computers, für den Spieler ein passendes Wort zu finden, gewinnt damit entscheidend an Bedeutung.

Beim Spiel Mensch gegen Mensch (bis zu vier Spielern) kann man natürlich in einer beliebigen Sprache spielen. Wörter, die dem Computer unbekannt sind, lassen sich nämlich zum vorhandenen Wörterbuch ergänzen (optional). Bei der Umsetzung des Originals tat sich noch ein weiteres Problem auf: Darf der Mitspieler die eigenen Buchstaben sehen oder nicht? Dazu gibt es zwei Lösungen.

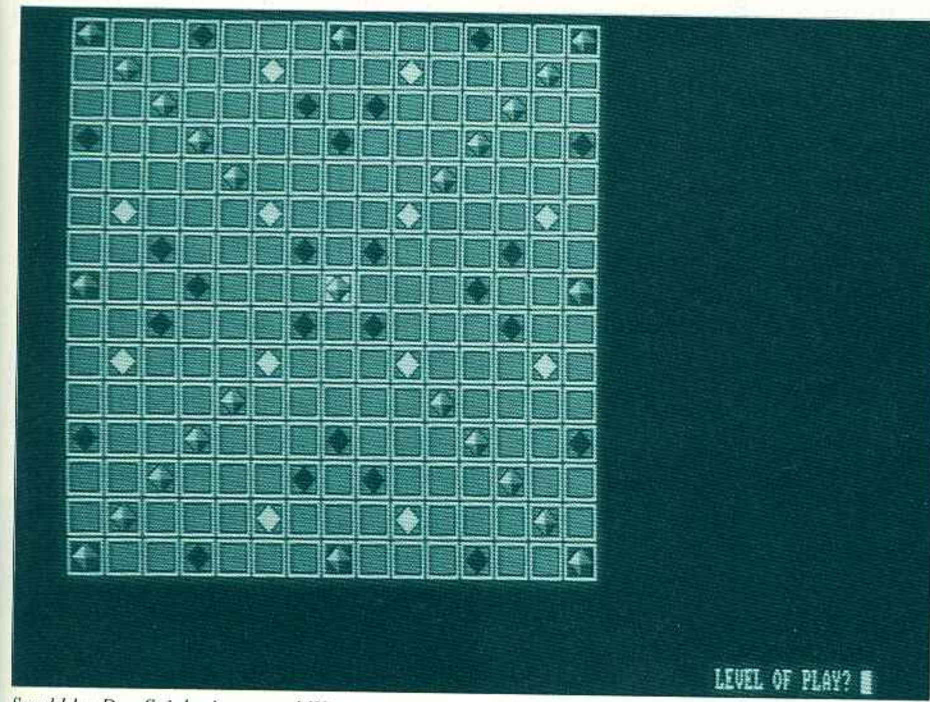


Trivial Pursuit reizt allein schon durch seine schöne Grafik

Mitspieler – In die Ecke

A): Der Mitspieler darf diese, wie auch beim Original, nicht sehen. Praktisch erscheint auf dem Bildschirm also in diesem Fall nur das Spielbrett mit den bis jetzt gelegten Wörtern und die eigenen sieben Buchstaben. Der Mitspieler muß während dieser Zeit pausieren, so daß er auf den Bildschirm keinen Einblick hat. Auf die Dauer erscheint mir diese Methode sehr unbequem. Vorzuziehen ist daher die Lösung

B): Sämtliche Buchstaben der Mitspieler werden auf dem Bildschirm gleichzeitig preisgegeben. Die spielfreien Mitstreiter können in ihrer freien Zeit dann schon nach neuen Buchstabenkombinationen suchen, so daß das Spiel auf diese Weise sicherlich nicht so langweilig wird, da niemand in einer Ecke des Zimmers auf seinen Einsatz warten muß. Jeder ist also während der ganzen Spielzeit direkt am Geschehen beteiligt.



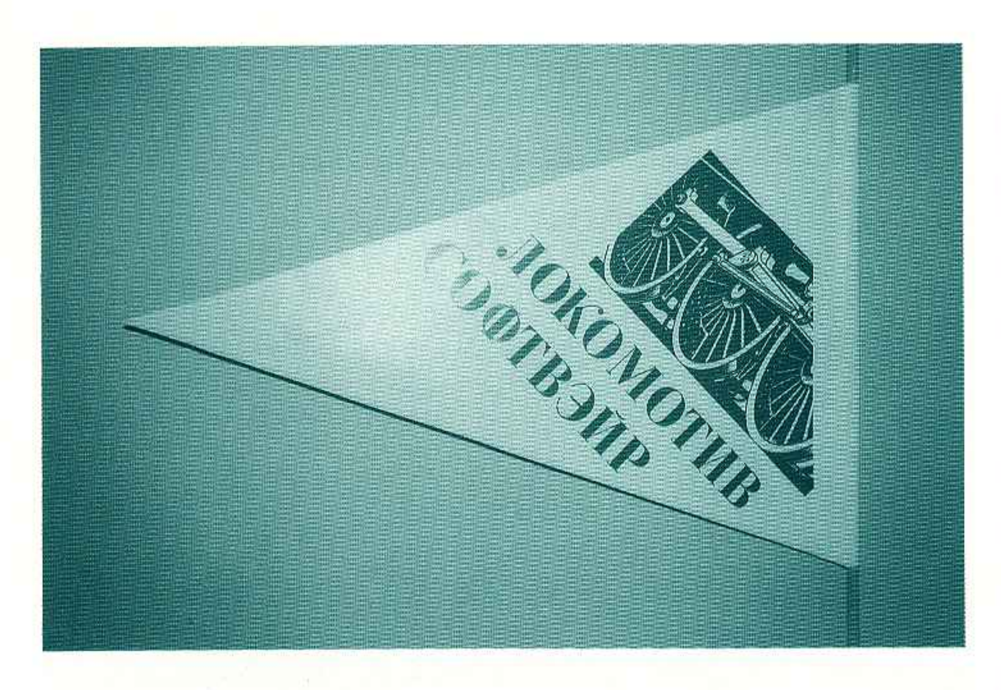
Scrabble: Das Spiel mit tausend Wörtern

Zu guter Letzt

Die Schlußbetrachtung: Für den englischen Markt ist Computer-Scrabble ohne Zweifel ein sinnvoller Zeitvertreib. Für den deutschen Markt ist es jedoch, trotz seiner guten grafischen Bildschirmführung, wegen des fehlenden deutschen Wörterbuchs und der oben genannten Einschränkungen nur schlecht zu gebrauchen.

Bernhard Graßhoff/rs

Die im Text angesprochenen Programme können Sie bei der Firma EDV-Obermeier, Bünder Straße 20, 4972 Löhne, zu den genannten Preisen beziehen.



LocoScript die 2.3te

Noch mehr Leistung für weniger Geld

LocoScript, das bei jedem PCW im Lieferumfang enthalten ist, wartet nun mit einer neuen Version auf. Zusätzlich gibt es ein neues Zubehörpaket, das für die Benutzung von Zweitdruckern mehr als interessant ist.

Wer sich einen PCW anschafft, wird sicherlich als erstes über die Textverarbeitung LocoScript stolpern. Da das Programm bei der achten Serie nur in der Version 1.41 ausgeliefert wird, wird sicherlich bald der Gedanke nach einer neuen Textverarbeitung laut werden. Doch warum in die Ferne schweifen? LocoScript gibt es in einer wesentlich besseren und leistungsfähigeren Version.

Was gibts Neues?

Gleich nach dem Start der Installationssoftware wird es jedem geübten LocoScript-Updateler auffallen. Ja, die Installationsanweisungen auf dem Bildschirm sind nun endlich deutsch. Es wird also kein Wörterbuch mehr benötigt, um das Programm zu installieren. Wer bisher eher unerfahren ist, sollte sich auch die ebenfalls in Deutsch gehaltene Installationsanleitung zu Gemüte führen und dort so einiges an Tips und Kniffen entnehmen ...

Derjenige, der LocoScript in Verbindung mit der Rechtschreibprüfung LocoSpell betreibt, wird sicherlich die neue Möglichkeit dankbar annehmen, die Wörter eines Benutzerwörterbuches mit der Funktion *F1-Text einfügen* in eine geöffnete Textdatei zu übertragen. Ist dies geschehen, können alle Wörter noch einmal überprüft und gegebenenfalls gelöscht werden.

Auch bei LocoMail hat sich einiges getan. So kann mit $= der Index festgelegt werden, nach welchem dann die eingelesenen Daten sortiert werden. Als Index kann entweder ein in LocoFile erstellter Index gelten, oder aber ein Feldname.

Langsam wirds knapp

Wie schon bei der Vorgängerversion existiert in der Druckerdatei MATRIX.PRI kein Hauptzeichensatz mehr, was die Größe dieser Datei wesentlich schrumpfen läßt. Sehr vorteilhaft, wo doch ein voll ausgerüstetes LocoScript mit allen Zusatzprogrammen nur noch ein paar kByte an freiem Diskettenspeicher übrig läßt. Bei einem vorhandenen zweiten Laufwerk können dann die einzelnen Schriftdateien (MATRIX. #??) auf eine Diskette für dieses Laufwerk kopiert werden.

Der noch verbleibende Speicherplatz auf der Startdiskette wird jedoch noch weiter schrumpfen, wenn an dem PCW noch ein Zweitdrucker angeschlossen ist. Dann muß nämlich noch die sieben kByte große Datei INSTALL.DRV auf die Startdiskette. Schrift- und .PRI-Dateien für den externen Drucker können dann jedoch auch auf die zweite Diskette ausgelagert werden.

Im großen und ganzen kann man also davon ausgehen, daß LocoScript mit der größten Wahrscheinlichkeit ab der nächsten Version voll ausgerüstet mit dem Mail-, Spell-, und File-Programm nicht mehr auf eine einzige Startdiskette (A+B-Seite) paßt. Auch wird der Speicher im Laufwerk M: langsam knapp, wobei es hierfür jedoch schon Alternativlösungen wie diverse Aufrüstsätze gibt.

Printer Support

Wie schon gesagt, wurde die Druckerverwaltung bei LocoScript umgestrickt. Ebenso auch das Konzept bei der Vermarktung der Zusatzsoftware, die teilweise zum Arbeiten mit LocoScript und diversen Zweitdruckern nötig ist.

Für runde 130 DM kann noch das sogenannte Printer Support Pack gekauft werden. Es ermöglicht, aus einer Liste von über 400 Druckern seinen herauszusuchen und anhand der Installationshinweise in LocoScript zu integrieren. Weiterhin interessant für Benutzer eines PCW 9512 sind Treiberdateien für die unterschiedlichsten Typenräder, die es bisher für den PCW eigenen Drucker gibt. Für den Fall, daß es für ein spezielles Typenrad noch keinen Treiber gibt, sind neben zwei Programmen noch Tips zum Erstellen von eigenen Treibern gegeben.

Ganz nebenbei ist im Paket für alle LocoScript 2.xx-Besitzer noch ein Update auf die neueste Version vorhanden und dies für die Achter-als auch für die Neunerserie.

Wer auf seinem externen Drucker gerne eigene Zeichensätze verwirklichen möchte, kann dies auch ohne Probleme tun. Hierzu gibt es das Programm LOCOCHAR.

Das Printer Support Pack beinhaltet also all die Funktionen, die zuvor in den Zusatzpaketen "Printer Character Sets Disc", "Printer Driver Disc" und "24 Pin Printer Drivers Disc" enthalten waren.

rs

Impressum

Herausgeber
Christian Widuch
Chefredakteur (verantw.)
Peter Schmitz (sz)
Redaktion
Jörg Gurowski (jg), Ralf Schössler-Niebergall (rs)
Autoren dieser Ausgabe
Norbert Finke, Gisbert Friege, Hannes Geyer, Bernhard Graßhoff, Heinz Hagemeyer, Christian Heinz, Holger Heyer, Friedrich Hinrichs, Wolfgang Horstmann, Andreas Knösel, Markus Kräutner, Christian Leubner, Andreas Lober, Clemens Lutz, Klaus Meffert, Ralf Möst, Damir Petkovic, Petr Potuznik, Ulrich Schmitz, Zbigniew Szkaradnic, Wolfgang J. Weber
Redaktions-Assistenz
Susanne Reckelkamm (re), Carmen Strube (cs)
Schlußredaktion
Christoph Schleef
Bereichsleitung
Matthias Bloß (Redaktion)
Uwe Siebert (Produktion)
Claudia Ebbrecht (Fotosatz/Lektorat)
Margarete Schenk, Helmut Skoupy (Montage/Reprografie)
Layout
Lars Völke
Fotografie
Klaus Jatho
Fotosatz
Reinhilde Schwarz, Heidemarie Kohlhaas
Montage/Reprografie
Monika Martin, Andrea Herschelmann, Susanne Eckhardt
Werbegestaltung
Mohamed Hawa
Anzeigenleitung
Wolfgang Brill
Anzeigenverkauf für PLZ 1, 4, 5
Gerlinde Rachow, Telefon: (05651) 809390
Sylvia Stephani, Telefon: (05651) 809380
Karina Ehrlich, Telefon: (05651) 809371
Bernd Heckmann, Telefon: (05651) 809381
Telefax: (05651) 809444
Anzeigenverkauf für PLZ 2 + 3
DMV-Verlagsbüro Hamburg
Kunaustraße 4a, 2000 Hamburg 65
Leitung: Sylvia Ehrenpfordt
Anzeigenverkauf: Sabine Bindseil
Telefon: (040) 6003075-77, Telefax: (040) 6015475
Anzeigenverkauf für PLZ 6 – 8
DMV-Verlagsbüro München
Zaunkönigweg 2c, 8000 München 82
Telefon: (089) 4391087, Telefax: (089) 4391080
Leitung: Britta Fiebig
Anzeigenverkauf: Peter Schätzle, Hannelore David, Ilona Sehm

Anzeigenverwaltung
Andrea Giese, Beate Kranz
Druckunterlagen – Disposition
Christina Wabra, Telefon: (05651) 809372
Anschrift Verlag/Redaktion:
DMV Daten- und Medien-Verlag
Widuch GmbH & Co. KG
Fuldaer Straße 6
3440 Eschwege
Telefon: (05651) 809–0
Telefax: (05651) 809333
Vertrieb
Verlagsunion Erich Pabel – Arthur Moewig KG (VPM)
Friedrich-Bergius-Straße 20
6200 Wiesbaden
Druck
Druckerei Jungfer, 3420 Herzberg
Bezugspreise
»CPC Amstrad International« erscheint zweimonatlich.
Einzelpreis DM 6,–/sfr. 6,–/öS 50,–
Abonnementpreise
Die Preise verstehen sich grundsätzlich einschließlich Porto und Verpackung.
Inland:
12 Ausgaben: DM 66,–
6 Ausgaben: DM 33,–
Europäisches Ausland:
12 Ausgaben: DM 96,–
6 Ausgaben: DM 48,–
Außereuropäisches Ausland:
12 Ausgaben: DM 120,–
6 Ausgaben: DM 60,–
Bankverbindungen:
Postscheck Frankfurt/M: Kto.-Nr.: 23043 – 608
Raiffeisenbank Eschwege:
BLZ: 52260385, Kto.-Nr.: 245 7008
Die Abonnementbestellung kann innerhalb einer Woche nach Auftrag beim DMV-Verlag, Postfach 250, 3440 Eschwege, schriftlich widerrufen werden. Zur Wahrung der Frist reicht der Poststempel. Das Abonnement verlängert sich automatisch um 6 bzw. 12 Ausgaben, wenn es nicht mindestens 6 Wochen vor Ablauf beim Verlag schriftlich gekündigt wird. Für unverlangt eingesandte Manuskripte und Datenträger sowie Fotos übernimmt der Verlag keine Haftung. Die Zustimmung zum Abdruck wird vorausgesetzt.



Noten lernen und Tonleitern nachvollziehen: ein Vergnügen mit dem "Musik-Tutor"

»CPC International« 2/3'92 erhalten Sie nur noch im **Bahnhofsbuchhandel** oder im **Abonnement** ab:



CPC

Eine Art "programmierten Musiklehrer" präsentieren wir Ihnen im nächsten Heft. Möchten Sie lernen, Noten zu schreiben? Ist Ihnen die Klaviertastatur bislang ein Buch mit sieben Siegeln? Oder wollen Sie vielleicht trainieren, beim Singen zuverlässig den richtigen Ton zu treffen? Der "Musik-Tutor" ist ein didaktisch gestaltetes Programm für den CPC, das Ihnen zu einem schnelleren Lernerfolg verhilft.

– "Datenfernübertragung" ist zwar heute kein Zauberwort mehr, das verschwörerische Hacker-Atmosphäre und geheimnisvolle Datennetz-Gruselgefühle à la "Big Brother" erzeugen könnte. Dennoch ist der Austausch von Daten über die "Strippe" im Zeitalter der Mailbox-Verbundnetze aktueller denn je. Daß auch CPC- und PCW-Benutzer nicht außen vor bleiben müssen, zeigen wir mit einer Artikelreihe, die ein erfahrener "Datenreisender" für uns schreibt. Wetten, daß auch Ihr Computer voll DFÜ-tauglich ist?

– Das Thema "Hardware" steht laut Umfrage bei unseren Lesern ganz hoch im Kurs. "Also gut", haben wir uns gedacht und einen Elektronik-Fachmann auf unsere geliebten Amstrad-Geräte losgelassen. Beim Auseinandernehmen der Peripherie ist dann so manche wichtige Beobachtung und so mancher nützliche Tip abgefallen. Lassen Sie sich überraschen!

Aktionen

– Treuen Lesern dürfte aufgefallen sein, daß die Rubrik "Personality" in dieser Ausgabe Pause hatte. Im nächsten Heft gibt es aber wieder einen Bericht über interessante Leute aus der 8-Bit-Szene.

– In Ausgabe 8/9'91 haben wir unsere Leser zu einigen Themen befragt. Die wichtigsten Ergebnisse lesen Sie im kommenden Heft. Dort erfahren Sie dann auch, wer die Preise im Gesamtwert von ca. 1800,- DM gewonnen hat.

– Der große Grafikwettbewerb ist beendet. Die schönsten Monochrom-Grafiken unserer Leser sehen Sie ebenfalls in Heft 2/3'92.

PCW

– Mit "Micro Design 3" kommt die taufrische neue Version eines außergewöhnlichen grafischen Gestaltungsprogramms auf den PCW. Funktionen, die die Grenze zum Desktop Publishing überschreiten, zeigen, was auch auf diesen Rechnern machbar ist.

– Daß der Joyce/PCW kein einsames Inselchen für versponnene Text-Eremiten ist, beweist einmal mehr die "APL/Z"-Implementation für diesen 8-Bit-Computer von Amstrad: eine klassische Programmiersprache aus dem Großrechnerbereich, die für anspruchsvolle arithmetische Aufgaben wie beispielsweise Vektorberechnung wie geschaffen ist.

– Hübsche große Schriften unter LocoScript hat sich mancher PCW-Benutzer schon lange gewünscht. Ein neues Software-Produkt aus England ist nun angetreten, diesen Wunsch zu erfüllen. "Endlich weg von den Standard-Schriftarten" – wir sagen Ihnen, ob diese Hoffnung gerechtfertigt ist.

DIE INSERENTEN

Com Zu Elektronik – Michael Kürbis..71
Crusader Software..........85
DMV..........2, 53 – 58, 63 – 68, 119, 120
Dobbertin..........29
G + L electronic..........85
Graßhoff..........71
Weeske..........8, 9
Wiedmann..........29



PCW DATABOX

DATABOX
ist der preiswerte Software-Service Ihrer Zeitschrift
DATABOX
enthält lauffähige Programme für Ihren Einsatz
DATABOX
lohnt sich auf jeden Fall - Monat für Monat

PCW 8256/8512/9512

Alles in Ordnung?
Sollten Sie vorhaben eine Dateiverwaltung zu schreiben oder zu verbessern? Benutzen Sie doch einfach einen der im Quelltext vorhandenen Sortieralgorithmen!

Grafiksystem selbstgemacht
Die aus Platzgründen nicht im Heft abgedruckten Programme zur Steuerung von Bildschirm, Drucker und Plotter bieten Ihnen die Möglichkeit, eine eigene Grafikverwaltung unter Turbo-Pascal einzurichten.

Inline für Z80
Dieses Pascal-Programm gibt Ihnen Auskunft über den aktuellen Inhalt der Register des PCW.

Bounce
Für alle Spielefreunde können wir das Programm Bounce mit über 100 Leveln anbieten.

Im Herzen des Joyce
Alle Beispielprogramme zum Artikel können Sie zum einen im Quelltext und ebenfalls startfähig compiliert wiederfinden.

Ein Hauch von Lexus
Zum dritten Mal können wir Ihnen unser Lexikonprogramm mit wesentlich verbesserten Funktionen auf der DATABOX anbieten. Zur Demonstration finden Sie ebenfalls eine Datendatei, die ein fruchtiges Geheimnis in sich birgt.

Einzelbezugspreis für DATABOX:
PCW – 3-Zoll-Diskette

24,– DM
Wenn Sie über den DMV-Verlag bestellen, gilt folgendes:

Inland:		Ausland:	
Einzelpreis	24,– DM	Einzelpreis	24,– DM
zzgl. Versandkosten	4,– DM	zzgl. Versandkosten	6,– DM
Endpreis	**28,– DM**	**Endpreis**	**30,– DM**

Zahlungshinweise:
Am einfachsten per Vorkasse (Verrechnungsscheck) oder als Nachnahme zuzüglich der Nachnahmegebühr. (Bei Lieferungen ins Ausland ist Nachnahme nicht möglich.)

Bitte benutzen Sie die Bestellkarte.

DMV-Verlag · Postfach 250 · 3440 Eschwege



Inhalt:

für den privaten Haushalt:
- Kontenverwaltung
- Haushaltsbilanzen
- Orakel: Hochrechnung effektiver Verbraucherkosten
- Lohnsteuerberechnung
- KFZ-Kosten
- Belegverwaltung
- Bankformulare
- Wertpapierverwaltung
- Komfortable Währungsumrechnung
- Tilgung
- Komplette Mini-Tabellenkalkulation

für kleinere Business-Anwendungen:
- Rechnungsschreibung
- Pecunia
- Bestellung und Bestand
- Superplan

Der Finanz-Fachmann

Ein CPC-Mega-Sampler, voll mit praktischen Anwenderprogrammen rund ums liebe Geld



DM 49,-*



Lernen mit Spaß

Sicherlich liegt es Ihnen auch am Herzen, daß die Sprößlinge eifrig und effektiv lernen und nicht nur stur büffeln.
Der Pädagoge Berthold Freier hat mit den Programmdisketten "Lernen mit Spaß" 1 und 2 ein Lerninstrument geschaffen, das in dieser Art einmalig ist. Auf den Disketten befinden sich zahlreiche Spielprogramme, die Ihren Kindern spielerisch das Verständnis für Zahlen, Text und Farben vermitteln. Auch das Allgemeinwissen wird ausreichend gefördert.
Für Kinder von 4 bis 12 Jahren wird durch "Lernen mit Spaß" 1 und 2 der CPC-Computer zum lehrreichen Freund.
Machen Sie Ihren Kindern mit "Lernen mit Spaß" eine Freude.

Lernen mit Spaß 1 **DM 24,95*** **Lernen mit Spaß 2** **DM 24,95***

Joyce-Highlights: Software-Perlen für Ihren Joyce/PCW

JOYCE-Highlights I: "Anwender"

Viele nützliche Arbeitserleichterungen

Spooler: Druckerspooler (256 und 512 KByte) • CAT: zeigt Disketteninhalt sortiert, auch versteckte Dateien • Loco2Pro: Konvertierung von LocoScript nach Prowort File • Rescue: Retten gelöschter Daten • Akte: Etikettenprogramm • Kalender: Feiertage, persönliche Daten • Drucken List V2.1: Listingdrucken und Kommentare • Funktionstasten: zeigt die Belegung auf allen fünf Ebenen • Für immer und ewig: Komprimieren und Entpacken von Dateien • Vario: Hardcopy auf (fast) allen Druckern • JOYCE-Tools: Benutzeroberfläche für Dateibearbeitung

nur DM 29,-*

JOYCE-Highlights II: "Spiele"

Für jeden etwas dabei. 14 ausgesuchte Spiele als Programmpaket – vom Kinderspiel über Unterhaltung bis zum kniffligen Denkspiel

- Gobang
- Pingo
- Backgammon
- Mensch, ärgere dich nicht
- Lokomotive
- Rubik's Clock
- Industriemanager
- Jackpot
- Poker
- Fruit
- Master Mind
- Ölmanager
- Verbindungen
- Q-Bert

nur DM 29,-*


Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland 4,— DM bzw. für das Ausland 6,— DM Porto und Verpackung. Bitte benutzen Sie die Bestellkarte.

DMV-Verlag • Postfach 250 • W-3440 Eschwege

